Contents

A User's Manual for

THE HYDROLOGIC SIMULATION ENGINE

Updated 05/21/97

Chapter 1

Introduction

The South Florida Regional Simulation Model (SFRSM) consists of two computer modules commonly referred to as the Hydrologic Simulation Engine (HSE) and the Management Simulation Engine (MSE). The HSE is capable of simulating one and two dimensional overland and ground water flow. The current document describes some of the ideas behind the conceptualization of the HSE and the development of the structure of the code. This document is somewhat common to the final C++ version of the code, and its predecessor, the FORTRAN version. The FORTRAN version was developed as an experimental tool to test the numerical methods, and to plan the design of the C++ version. The C++ version may look much different from the FORTRAN version in the end; but, the basic concepts may not be much different from what is described in the document. A description of the FORTRAN code is also included in this document. Governing equations and the mathematical derivations are described in a separate document.

The HSE consists of a two-dimensional overland flow module, a two-dimensional ground water module, and a one-dimensional canal flow module. The overland flow module and the canal flow module use diffusion flow instead of complete depth averaged dynamic flow described by the Saint Venant equations. The approximation is justified because the inertia terms are extremely small in the study area. A semi-implicit finite volume method is used to solve the 2-D diffusion flow equations. The flow domain is discretized using an un-structured triangular grid. The model is developed so that it can be used with almost any time step, and almost any triangular discretization used in practical applications. A weighted implicit method is used to improved stability. This model, as in the case of any other model does not have a guarantee against failure due to nonlinear instability. The model can simulate structures, levees, and various other boundary conditions.

The 1-D canal network model also uses a finite volume method in the 1-D sense. Any network of canals with practically any number of boundary conditions can be solved using the 1-D model. The overland flow, ground water flow and the 1-D canal flow modules can be used either as components of the HSE, or independently as free-standing models. The modules can be used to make short or long term simulations over large or small areas when diffusion flow approximation of the St. Venant equations is valid. During the solution using the implicit method, the overland and groundwater modules populate sub-matrices that will finally be sent to the sparse solver. The canal flow module populates another sub-matrix. Some of the remaining elements of the matrix get filled up depending on the interactions between the overland and canal flow elements.

A numerical model is not exactly equal to the physical system it represents. It only approximates the hydraulic and other characteristics of the area. In the present case, triangles are used to represent natural areas, parking lots, farms and various other areas which are not so triangular. As a consequence, the results coming out of a model are only as good as the triangular approximation used to represent natural areas. Fortunately, minute features in natural areas have a little effect on the overall flow patterns that dominate an area, and therefore, a set of finite triangles in most cases can represent the flow domain sufficiently accurately. Minor geometrical disparities can be compensated by calibration of parameters assigned to the idealized triangles. Fortunately, physically based models such as the current model have been shown to be useful and adequate even if the model does not have the exact physical representations of local conditions.

The current documentation includes some basic ideas behind the conceptualization of the model, definitions of variables, and a descriptions of the test cases. The test cases are useful in verifying the accuracy of the models, demonstrating the use of the model, and setting up benchmark solutions so that the model can be run during different stages of development to check if the integrity of the computations is violated specially after modifications. Some intermediate results are also included to assist in isolating any bugs. A benchmark test for run time will also be established so that the model run time can be compared after a model enhancement.

The documentation also includes some theoretical formulations useful in either developing the model, or justifying its formulation, not having a clearly defined location in any other documentation. The section on the "mathematics of the source term" is one such example. These sections will be removed as soon as other vehicles for the purpose become available.

1.1 Ideas for the future

This is one section that is very difficult to write for a computer model because everything about modeling is changing. However, it is useful to document the prevailing wisdom so that alternative ideas can be attracted in the future without hinderance. Two of the planned ideas are the introduction of parallel processing, and the periodic updating of the sparse solvers to make sure that the model used the best available computer technology.

Considering the immediate modeling needs, a water quality model using a simple transport algorithm is not an extremely far fetched idea. The flexible architecture of the model supports an easy implementation, and possible expansion into a more accurate module with ecological, fire or other plug-ins.

If improvement of the diffusion network model is considered to be important, coupling of a dynamic river network model using complete equations to the same sparse solver is one suggestion. Such a modification will enable the use of full equations in selected areas, and maintain the extremely efficient diffusion modeling over other areas.

In terms of algorithm developments, a full equation model is always a possibility. Finite element models that can carry out this task are already available, even if the efficiency of such models in a typical application to South Florida is low. One of the recent methods that drew the attention is a second order projection method already tested with Navier Stokes Equations by NASA (Lou, 1997). This is a splitting method that can also be used with the current diffusion solver to improve its velocity solution. Starting with the diffusion solution at every time step, a hyperbolic solver module can be used to improve the approximate velocity solution obtained by the diffusion solver.

Chapter 2

Two-dimensional overland and groundwater flow models

The application of the 2-D overland and groundwater flow models to arbitrary regions requires discretization of the model domain into triangular cells. The hydraulic characteristics of the region are then assigned to the cells. Flow conditions in the model are explained using water levels of the cells, and flows across walls. Boundary conditions are assigned to cells or cell walls. If flows across specific flow lines are needed as model output, the model uses cell walls to compute and monitor them.

In the current finite volume formulation using triangular cells, circumcenters are used to represent the respective triangles. This is due to the mixed finite element method that underlies the current derivation of the finite volume method. Water level, ET, roughness, or any other cell variable at the circumcenter may be slightly different from its value at the centroid, which truly represents the triangle. However, considering the complexity of transferring values between the circumcenter and the centroid, and considering the need for consistency, cell average values are used in the computations throughout the simulation to represent cell values. Circumcenters are used to represent the geometric locations of these average values. Several numerical experiments were carried out to find a way to transfer values between the centroid and the circumcenter using the average water surface slope of the triangle. However, it was found that the error in the adjustment was too large at times, and the method failed. Numerical experiments showed that the results are accurate even without these corrections. If this adjustment is found to be important in the future, studies may be necessary to develop a better method, or the distance between the two points may have to be kept significantly small by selecting triangles that are not extreme in shape with respect to this criterion.

2.1 Discretization of the model domain

Implementation of the finite volume method requires the model domain to be discretized into cells in the shape of polygons. The current implementation however limits the polygon shape to triangles. Furthermore, model errors are found to be smaller with acute angled triangles. The triangular discretization method used in the model is popular among most finite element models. These triangles are expected to completely cover the model domain. The numbering method for triangles is the same used by grid generation packages such as the Argus mesh maker, GMS and TECPLOT.

Finite volume triangles are marked using cell numbers $1, 2, \ldots ne$ in which ne is the number of cells. Variables such as the water level, ground elevation, ET, rainfall, and parameters such as flow roughness are defined for cells. Nodes or vertices of the triangles are numbered using node numbers. An input data file describing a tessellation must consists of nodal connectivity and nodal coordinate data. Table 2.1 shows a sample data set for the area in Fig. 2.1. The connectivity of cells is described using node numbers around them written in clockwise or counterclockwise direction. A clockwise direction is used in the example. In Table 2.1, cell 1 is defined by nodes 1, 2 and 6, and the data line simply has 1 1 2 6. In the example, the origin is assumed to be at node 13, and the length of a small square is assumed to be 5000.0 m. The order of numbering of cells or nodes does not affect the run time of the model because of the use of sparse solvers instead of banded matrix solvers. If nodes and elements are numbered with gaps, internal representation of the code has to skip the gaps to create the consecutive numbering system used within the model. Most mesh generators are capable of creating connectivity and nodal coordinate data that can be used directly in the model.

An ideal discretization of a domain will have equal sized equilateral triangles filling up the space with no jaggerred edged boundaries. Triangles closer to equilateral in shape give smaller errors. Obtuse angled triangles may produce larger errors. Without preconditioners, the condition number of the solution matrix may be proportional to the ratio of the largest to the smallest cell areas under uniform hydraulic conditions. A closer to ideal discretization would reduce both the numerical errors in the model and the number of iterations, by creating well conditioned stiffness matrices. Uniformly sized triangles have the same advantage in explicit schemes too because stability is decided by the size of the smallest triangles.

A model domain described using cells is bounded by cell walls. Cells contain water, and cell walls contain functions that control the flow of water. Rainfall, infiltration, percolation, evapo-transpiration, containment of unsaturated flow, and other functions mainly associated with the vertical movement of water are described using functions and variables defined for cells. Two dimensional flow, structure flow, groundwater flow and other functions associated with the horizontal movement of water are described using variables defined for walls. Cells and walls are numbered in the model. The internal representation of cell walls is important to the user only because some internal and boundary conditions are associated with them. Levees, structures and canals are some of the flow functions attached to walls.

Table 2.1: The data set describing the discretization in Fig. 2.1.

CONNECTIVITY

CELL	NODES		
1	1	2	6
2	2	3	7
3	3	4	8
4	5	6	10
5	6	7	11
6	7	8	12
7	9	10	14
8	10	11	15
9	11	12	16
10	1	6	5
11	2	7	6
12	3	8	7
13	5	10	9

....

NODAL CO-ORDINATES

NODE	X	Y
1	0.0	15000.0
2	5000.0	15000.0
3	10000.0	15000.0
4	15000.0	15000.0
5	0.0	10000.0
6	5000.0	10000.0
7	10000.0	10000.0
8	15000.0	10000.0
9	0.0	5000.0
10	5000.0	5000.0
11	10000.0	10000.0

....

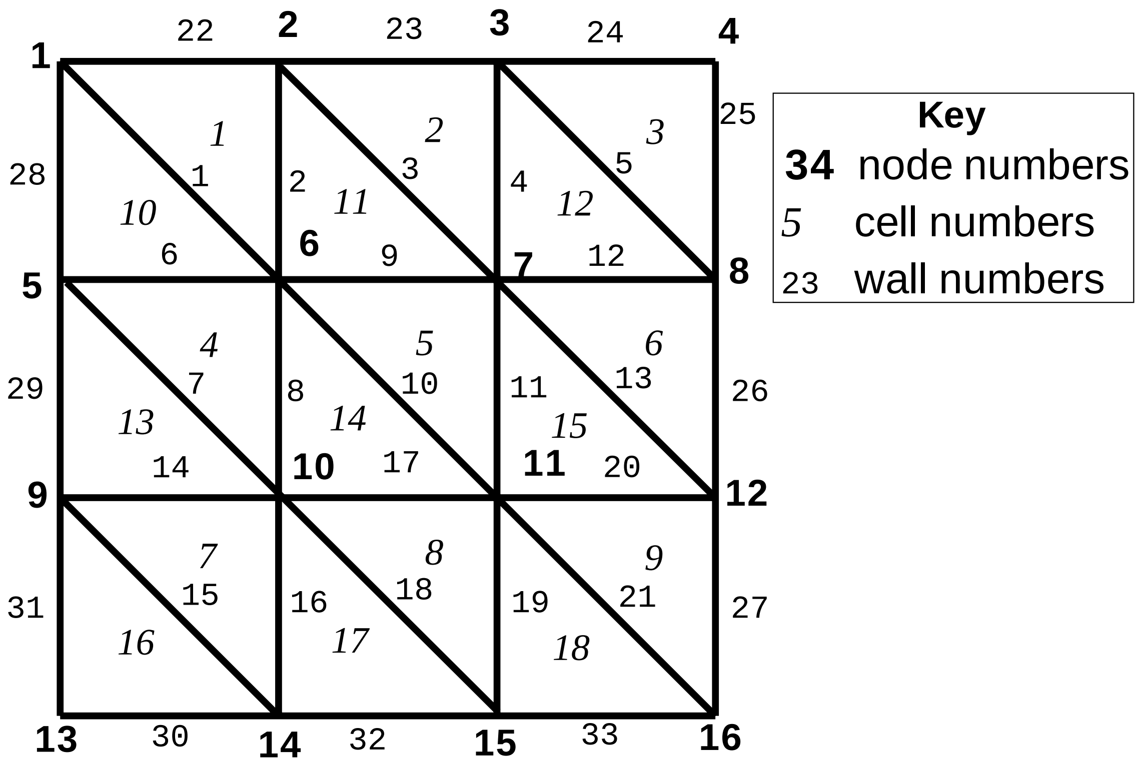

Figure 2.1: A finite volume grid discretization.

2.1.1 Cell walls

All the walls are internally numbered by the model, and the user has to know them only to assign them to structures and other flow features. By default, all the internal walls are assigned as 2-D flow walls, and all external walls are assigned as no-flow walls. All the walls sharing a node with the boundary are assigned as 1-D walls. Any of the default type can be changed by defining the required function in the code, and associating it with the data in the input data file. Walls are numbered arbitrarily without gaps in the numbering. Table 2.2 shows how data is represented internally in the model for the example in Fig. 2.1.

Column 1 of Table 2.2 shows the wall numbers. Columns 2 and 3 show the cells on opposite sides of the wall. In the case of external walls, only one cell is found, and a value of 0 is assigned to the second place. Letters N1, N2, N3, N4 are also used in the table to explain the data. Columns marked N3 and N4 show the node numbers defining the wall,

Table 2.2: An example describing the cell walls.

Wall	Cell N1	Cell N2	Node N3	Node N4	OL Type	OL Seq.	GW Type	GW Seq.
1	1	10	1	6	0	0	0	0
2	1	11	2	6	1	0	1	0
3	2	11	2	7	0	0	0	0
4	2	12	7	2	1	0	1	0
5	3	12	3	8	4	1	3	0
6	4	10	5	6	4	2	0	0
7	4	13	5	10	2	0	2	0
...								

Overland flow wall Types:

0 = No flow wall

1 = Overland flow type wall

2 = Direct cell to cell flow based on Manning's equation.

3 = Uniform flow

4 = Structure type A (whatever)

5 = Structure type B (whatever)

6 = Weir used for the Kissimmee study.

7 and over... available ..

Groundwater flow wall types:

0 = No flow wall

1 = Laplacian flow walls

2 = Direct cell to cell flow based on Darcy's equation.

3 = Canal seepage walls (not defined yet)

automatically computed by the model pre-processor. N3 and N4 are placed in ascending order. The ascending order is used so that the wall can be searched easily when trying to assign boundary conditions. Column 6 shows the type of the wall, which decides the type of flow transfer that can take place across the wall. If the overland (OL) flow type is 0, it is a no-flow wall. The bottom section of Table 2.2 shows various overland and groundwater flow walls used in the FORTRAN code.

2.1.2 Flow conditions associated with walls

Flow functions and certain boundary condition types such as the no-flow type and the structure type are defined at walls. These boundary conditions are specified by changing the default wall types of 1 for internal walls, 0 for no-flow walls and 2 for 1-D walls to the new types. In the internal representation shown in Table 2.2, walls 5 and 6 for example are for structures. The input data file provides the necessary boundary condition information to fill this internal data. A sample boundary condition file is shown in Table 2.3. The first row of the table shows a structure type 4 spanning between nodes 3 and 8. The sequence numbers 1 and 2 indicate that there are two structures of the same type 4 using different sets of parameters. The parameters used for the structure equations are different in this case even if the same function for structure type 4 is called for both structures. The parameter values can be listed immediately below the line in the data file, or in a different data file. The values 0.012, 25.3 etc. in the example are the parameters for the first structure of type 4. The values 0.012, 13.2, etc., are for the second structure of type 4. The parameters may include weir coefficients, gate openings or other relevant data. Row 5 of Table 2.2 shows a structure type 0 which represents no flow as in the case of a levee. Once values in the table are read, the default wall types are changed to create Table 2.2.

Table 2.3: An example of an input overland flow boundary condition file.

```
Node N3     Node N4    Type    Sequence
```

3		8	4	1
0.012	25.3	2.2E-3		
5		6	4	2
0.012	13.2	2.1E-3		
1		6	0	0

Boundary conditions associated with ground water flow are assigned in a similar manner, and can be read from a similar data file. In the internal representation, wall 5 of Table 2.2 for example, the ground water wall is of type 3. This type can be defined within the code as needed. It can easily be a sheet pile wall with certain seepage characteristics. When the ground water flow boundary condition file is read, the default ground water wall types which are the same as overland flow wall types are changed to the new types. Wall and cell numbers and the nodal connectivities do not change as a result of boundary condition assignments.

2.1.3 Flow conditions associated with cells

The basic state variables in the model are the water levels. Water levels, ground elevations, and the parameters required to obtain the vertical solution are associated with cells. Rainfall, ET and infiltration are three of the quantities considered in the vertical solution, and are also associated with cells. Other parameters associated with the cells include the roughness coefficients, hydraulic conductivities, and the storage coefficients. Boundary flow and well pumping are also associated with cells, and are treated similar to the way any other source terms are treated.

When there is a pumping well or a flow boundary condition located in a cell, the correct flow rate is taken into or out of the cell based on the flow time series data file. The location of the well within the cell or the identity of the wall with the flow boundary condition are not important for the computations. Table 2.4 shows the representation of inflow or outflow into cells 3, 4, 5 and 8. Inflows into cells 3 and 4 are associated with input time series 2, and the inflows into cells 5 and 8 are associated with input time series 1 and 3. Even when the inflow data is associated with certain walls as in the case of flows through external walls, the net effect on the model is the gain or loss of flow in the cell. Therefore, wall numbers are not used when specifying flow boundary conditions. When time series data for a boundary are read, they are related to the correct cells using sequence numbers shown in Table 2.4.

Table 2.4: An example of inflow boundary condition data

CELL NO.	SEQUENCE NO.
3	2
4	2
5	1
8	3

2.1.4 Head boundary conditions

It is possible to assign the water level of any cell to an input time series. The data required includes the cell number, and the sequence number of the time series as shown in Table 2.5. The time series data can be provided at regular time intervals which can be different from the length of the time step. The value at any time within the regular time interval is computed using linear interpolation.

Table 2.5: An example of a head boundary condition.

CELL NO.	SEQUENCE NO. OF TIME SERIES DATA
17	2
18	2
12	1
6	3

2.1.5 Simulation of flow in the vicinity of boundaries

When boundary conditions are specified at walls causing the replacement of two-dimensional flow walls with new walls, flows across any nearby walls sharing nodes with the new boundary wall are affected. This is specifically true in the case of no-flow boundary walls. The flow is then confined to be tangential to the no-flow wall, and the flow across neighboring walls cease to be fully two dimensional. This local affect is taken into account by converting the neighboring walls to "direct flow" or 1-D walls across which the flow rate depends on the distance between their centroids, water levels, and the average conductivity characteristics. The modification of the 2-D wall type to direct wall type is done internally.

A similar wall type modification is carried out during the drying of cells too. Dry areas temporarily create a no-flow wall between the dry and wet boundaries. These no-flow walls become fully 2-D walls when the cells become wet again.

2.1.6 Guidelines for discretizing 2-D domains

Hydrologic Models are tools used to understand hydrologic conditions of landscapes. The results of models are only as good as the underlying concepts and the assumptions used in developing the models. Discretization is one of the processes during which many assumptions are made about the shape of the area being modeled. Model developers make every effort to develop models that can be accurate under as many conditions as possible. However, there are many practical and resource limitations that make it difficult to achieve this goal. Even if every attempt is made to prevent disasters, it is easy for any model to be used under a wrong set of conditions, and obtain useless results. Following guidelines direct the user to conditions under which the current model can be safely used. The model is designed such that the governing equations and the assumptions behind model conceptualizations are not violated under most of these conditions. In many models, this

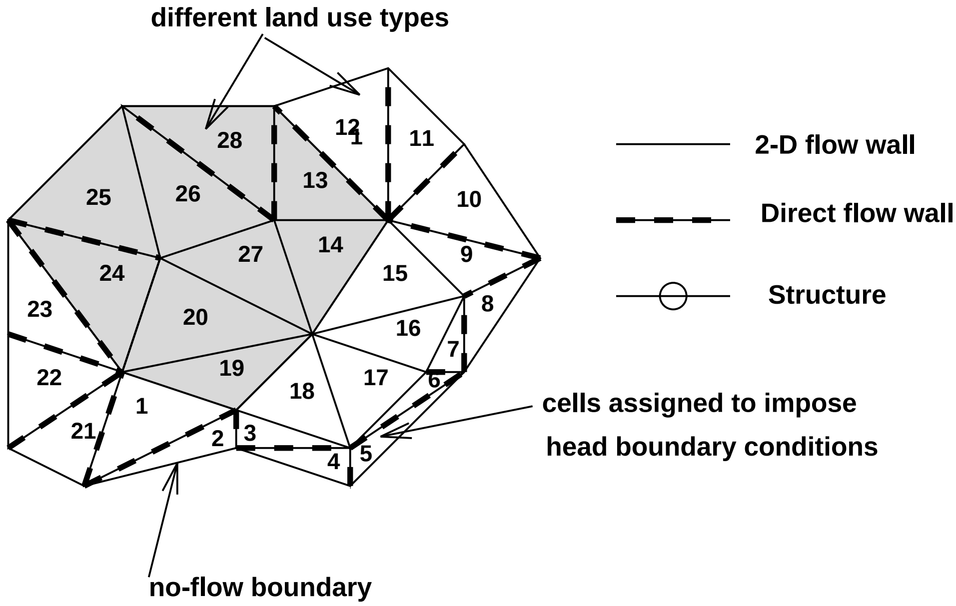

Figure 2.2: Definition sketch explaining cell and wall types.

is not an easy task. For example, covering the South Florida landscape with triangles is never possible if a very high level of perfection is expected, because natural landscapes rarely have linear boundaries. The accuracy of the model results depend on the accuracy of the representation itself. If the representation is really bad, it may not be advisable to expect much out of the models either. However, most reasonable representations have been found to be adequate for practical purposes, when used with caution. The following list contains some of the basic steps useful in creating meshes and making model runs.

1 *Mark the model boundaries:* The first step in the discretization involves marking the model domain. It is important to have boundaries which are easily definable from a hydraulic point of view, than to limit the model coverage to exactly what is needed. It may be necessary to include certain areas that are not ordinarily needed, to achieve this goal. Easily definable boundaries include no-flow boundaries as in the case of major flow divides, tidal or lake boundaries, structure boundaries with known structure characteristics, etc. Boundaries should not be placed across major

flow paths with unknown or inaccurate flows, except under extremely desperate circumstances.

2 *Mark land use type boundaries:* Within the areas marked for the model domain, areas designated to specific land use types have to be marked. It is ideal if triangles or polygons do not have more than one land use type. If mixed land use types become necessary, the physical properties have to be weighted to take mixed types into account.

3 *Mark areas which need more resolution:* If the topographic elevation or the land use type vary significantly, and if it is necessary to study local flow conditions in detail, it is important to have a higher resolutions. In such cases, the solution accuracy can be improved by introducing a gradual transitions of polygon shapes and sizes between high resolution areas to low resolution areas. A gradual transition can also prevent the triangles becoming too far from equilateral.

4 *Locate structures and levees:* Cell walls have to be placed at levees and structures so that they can be assigned as proper wall types. Structures associated with 1-D canal systems have to be considered separately.

5 *Mark boundary condition cells:* When the cells used to impose the head boundary conditions are large, head boundary conditions can introduce errors. Such errors can be minimized by refining the cells in the neighborhood of head boundaries. Cell refinement is useful near other kinds of boundary conditions too. In the case of flow boundaries, mesh refinement can introduce the correct length of flow path between the boundary point and the rest of the model domain.

Two dimensional flow may not occur at every point in a model domain. When one or more wall is connected to a no-flow type wall or a structure type wall, the model automatically changes the neighboring walls, if they are 2-D walls, to direct flow walls. The approximation is useful closer to no-flow and structure type walls because it is difficult to find representative nodal elevations in these cases. If the flow is known to be one-dimensional in a local area, this boundary condition can be assigned to model the flow.

After discretizing the space, data files have to be created to describe the properties of the generated cells. Except for the imposition of wall type boundary conditions, geometry data sets can be prepared using standard software packages or GIS utilities as described later.

2.2 Model input data

A major portion of the input data consist of geometry data which can be created using GMS or other software tools. Boundary condition data also can be considered as part of geometry data. In addition to geometry data, many other types of data also have to be prepared for a model run. They are referred to as primary input data, secondary input data, time series data and output control data for convenience.

2.2.1 Primary input data

Primary input data consist of the time step length, implicit weighing factor, solver option, and few other data values have to be selected before the model is run. This data is unique to a model, and can be easily explained without a subjective bias.

The accuracy and the run time of a model of a given discretization is decided by the length of the time step. A good reference for this behavior is described by Lal (1996). The information in the paper is relevant for both rectangular and triangular cells. When the overall cell size is reduced, the accuracy of the model increases, but the run time increases even by a larger proportion. Small time steps are useful in improving the accuracy of models, but may also increase run times. On the other extreme, very large time steps can reduce the model accuracy, and can even make the model unstable due to nonlinear instabilities. Some of the primary input variables are listed below.

Table 2.6: Definition of primary input data variables.

Name	description
TT	total simulation time
NT	number of time steps
ALP	weight used in the weighted implicit method; 0.0 is used for fully explicit problems and 1.0 is used for fully implicit problems.
METH	solver option.
IOPG	option to select ground water or overland flow type problems.
NB	number of wall type boundary conditions.
NE	number of cells
ND	number of nodes
NH	the number of head boundary conditions specified.
NITER	number of iterations used to refine the stiffness matrix.
NQ	number of flow boundary conditions.

2.2.2 Secondary input data

Tolerances and other parameters which cannot be easily explained physically, but are essential to make accurate and efficient model runs fall into this category. The values of these parameters are rarely changed except in the case of new model applications. This data is used to enhance the computational efficiency of models without sacrificing the accuracy of solutions. Following is a list of secondary input parameters, along with brief descriptions.

STOL:

This parameter is active only when the Manning's equation is used. When the water surface slope is small, a singularity is encountered when computing K because of the division by the slope term. To avoid this, Manning's equation is replaced with a linear equation when the water surface slope is smaller than a small tolerance, STOL.

If the STOL used is too small, the resulting K coefficients in the matrix can potentially make the matrix near-singular, and the solver may require too many iterations to converge. If STOL is too large, the solver may converge fast, but the solution may use an approximate equation rather than the Manning's equation more often, possibly affecting the results. The best value for a given model can be found after experimentation. Figure 2.3 shows how the number of iterations and the upstream head vary for one of the Kissimmee river applications. According to the figure, there is no advantage in selecting a STOL value much smaller that 1.0 $\times 10^{-5}$ and larger than 1.0$\times 10^{-2}$. Some experimentation indicated that the ratio of the smallest slope in the model to the largest slope in the model cannot be less than the machine precision times a numerical safety factor (10-1000). According to this guideline if the model is to be applied over a mountainous area, STOL may have to be increased to a larger value.

$ALPHA$

This parameter is used as a weighing factor in the weighted implicit formulation. Values of ALPHA of 0.0 and 1.0 changes the model from a fully explicit mode to a fully implicit mode. A value of 0.5 is suggested for many weighted implicit models because of the central time differencing and the associated higher order accuracy in time. In dynamic 1-D models such as MODBRANCH and UNET, $ALPHA$ values in the range of 0.6-0.8 are selected to improve model stability while maintain the accuracy. The current model shows the same behavior with ALPHA. A value of 1 gives the highest stability, and a value of 0.5 gives the highest accuracy. A value within this range is used for most model runs.

TOL

This parameter is closely associated with the solver. It controls the maximum allowable error norm before the iterations stop. The SLAP package assumes a value of 250.0 $\times$ machine precision for this parameter. For single precision, the machine precision is

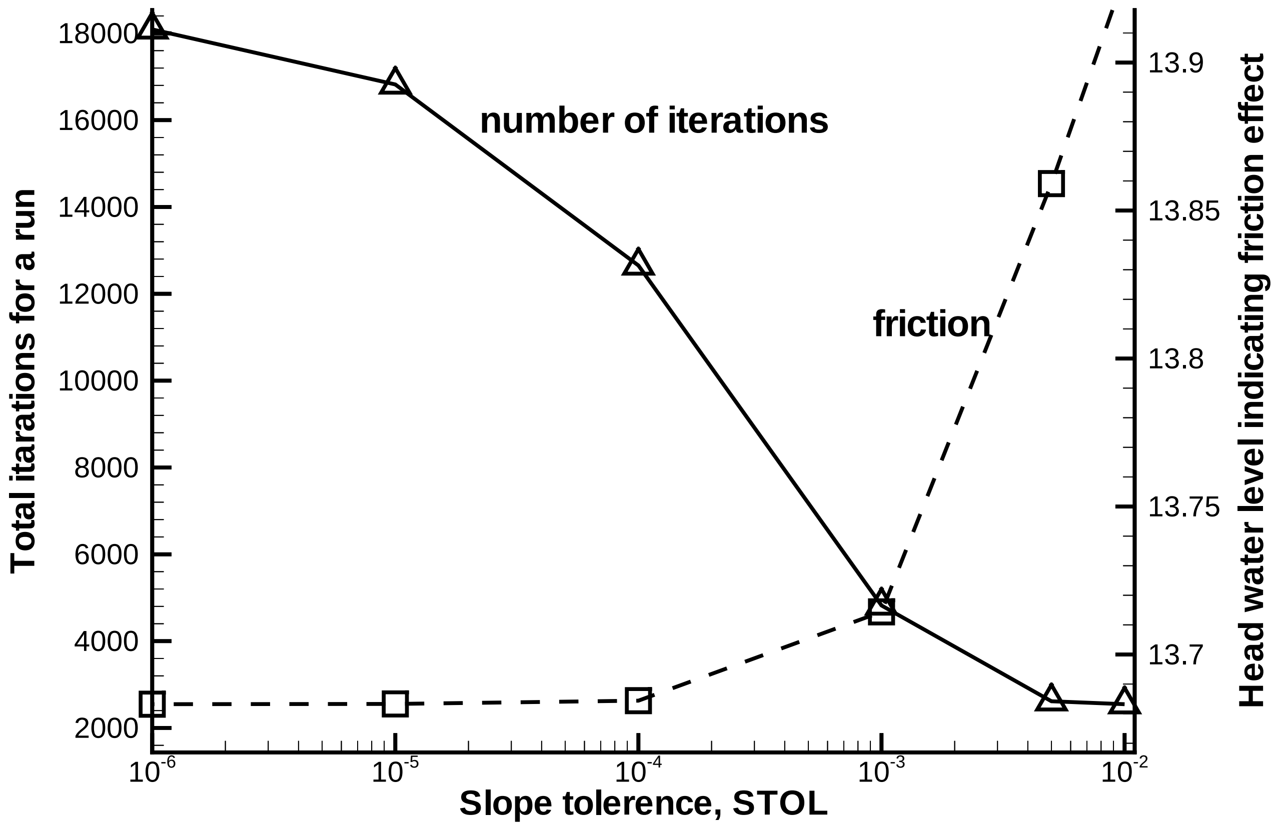

Figure 2.3: Figure showing the variation of the solution and the number of iterations with STOL.

approximately 1×10^{-7} and for double precision, it is 1×10^{-16}. The only time this tolerance has to be changed is when the number of iterations is extremely large. In that case, its value can be increased to 500.0 $\times$ the machine precision, according to the SLAP 2.0 documentation.

HHTOL

This parameter is used to control the wetting and drying of cells in the overland flow model when ground water is absent. A value of 0.001 m was used in the model, even if any other value larger than the machine precision is possible. When the difference between the water level and the ground level is less than this value, the ground is assumed to be dry. The purpose of the parameter is to prevent the model from computing diffusion flows when the depth is extremely small. Computation of flows with extremely small depths is time consuming, and inaccurate. When a detention depth is used in a model, it is used instead of HHTOL.

NITER

This parameter specifies the maximum number of iterations allowed within a diffusion flow time step. During most diffusion flow computations, the overland flow K value used is computed using model state variables from the previous time step. However, the true value of K may be different for the current time step, and its best estimate can be obtained by iteration within the same time step. If NITER is set to 0, no iterations are carried out, and the default method is used in which K values are obtained from the previous time step. Experimentation with the test problems showed that the number of iterations needed is very small.

METH

This integer is used to select the method to solve the system of linear equations.

2.2.3 Output control data

A typical model output consists of water levels and flow velocity vectors at selected locations. In addition, discharge rates across lines in the domain are also useful. Since there may be too many data points in a model run, a limited number of points are generally selected at which the output is printed. The data that controls the output consists of a list of cell numbers at which the stages are required, and details of the flow lines across which discharges are required.

2.2.4 Marking flow lines

Flow across certain arbitrary lines in the South Florida landscape have become important in decisionmaking. These flows can be computed by post-processing model output, or by assigning the flow line information as part of the model input so that the required flow can be added to the output. Flows across cell walls are computed by multiplying the wall K of the overland flow or the ground water flow by the head difference. Figure 2.4 shows part of the finite volume model grid, and a line across which the flow is monitored. Flow lines are given sequence numbers in the model. Each flow line is composed of a number of cell walls. Flow across flow lines are computed by algebraically adding the flows across each of the walls. The flow line in Fig. 2.4 is expressed in the data file as shown in Table 2.7. In the example, the flow through the walls in a North to South direction has to be computed. The walls are made of nodes 1-3, 3-4 and 4-5. The input data shows the two cells across which the flow takes places, with the first cell showing the donor cell, and the next cell showing the receiver cell. The output gives the algebraic sum of flow rates from cell 10 to cell 11 + cell 12 to cell 13 + cell 15 to cell 14 in a time series.

Table 2.7: Example of an input data file defining a flow line.

```
Number of walls = 3
N1 N2 (N1 is the donor, N2 is the receiver)
10 11
12 13
```

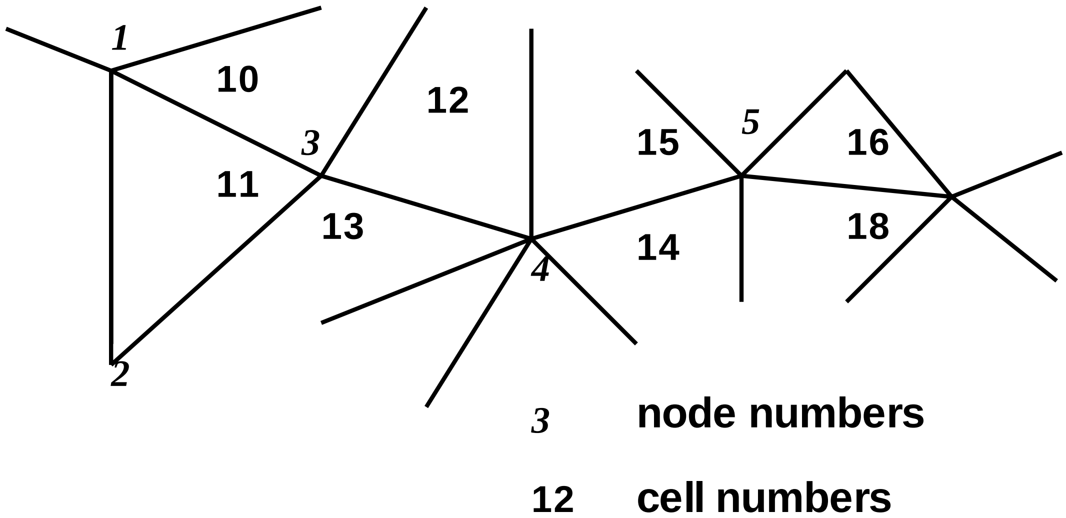

Figure 2.4: Definition of a flow line.

```
15 14
```

2.2.5 Time series data

Time series data made of sequential data values are used in the boundary conditions or the source terms of the governing equations. Boundary conditions require head or discharge time series. Source terms require mainly rainfalls and evapotranspirations. Every cell in the model gets the rainfall and the ET data from a time series data file. Every boundary condition too gets stage or flow information from a time series data file. Each of the boundary condition time series is numbered, and identified along with the boundary condition during the specification of the boundary condition. Time series data is provided at regular intervals which can be different from the run time steps.

2.3 Model variables

Both overland and groundwater models use the average cell water level as the state variable. The current and next time step values of this variable for each cell are H0(I) and H1(I), respectively. Discharges across the walls are computed using these water levels. Since the model solves only for the water levels, a model run can be completed without ever computing the discharges. Discharge is a variable associated with the walls. Model parameters related to land use types are assigned to cells. Manning's roughness, conductivity, and storage coefficient are three such parameters. Appendix C shows a list of variables used in defining the model.

2.4 Vertical solution

Previous sections described the computational module used to simulate the horizontal movement of water due to overland flow and groundwater flow. The vertical movement of water through the soil is considered in the vertical solution module. Rainfall, infiltration and evapotranspiration are considered only in the computation of vertical flow. Exchange of water between the cell walls is neglected during the computation of vertical flow. In formulating the model, it is assumed that the horizontal and vertical flows can be solved separately within each time step. If this assumption is not made, other simplifying assumptions have to be made to proceed with the solution.

Figure 2.5 shows a sequence of steps suggested to carry out the vertical solution. This sequence is based on the existing SFWMM. In the future, it may be necessary to experiment with a number of sequences before deciding the best suitable for the model. All sequences should ideally converge to the actual solution when the time step is made small. The best sequence should give accurate results even with large time steps.

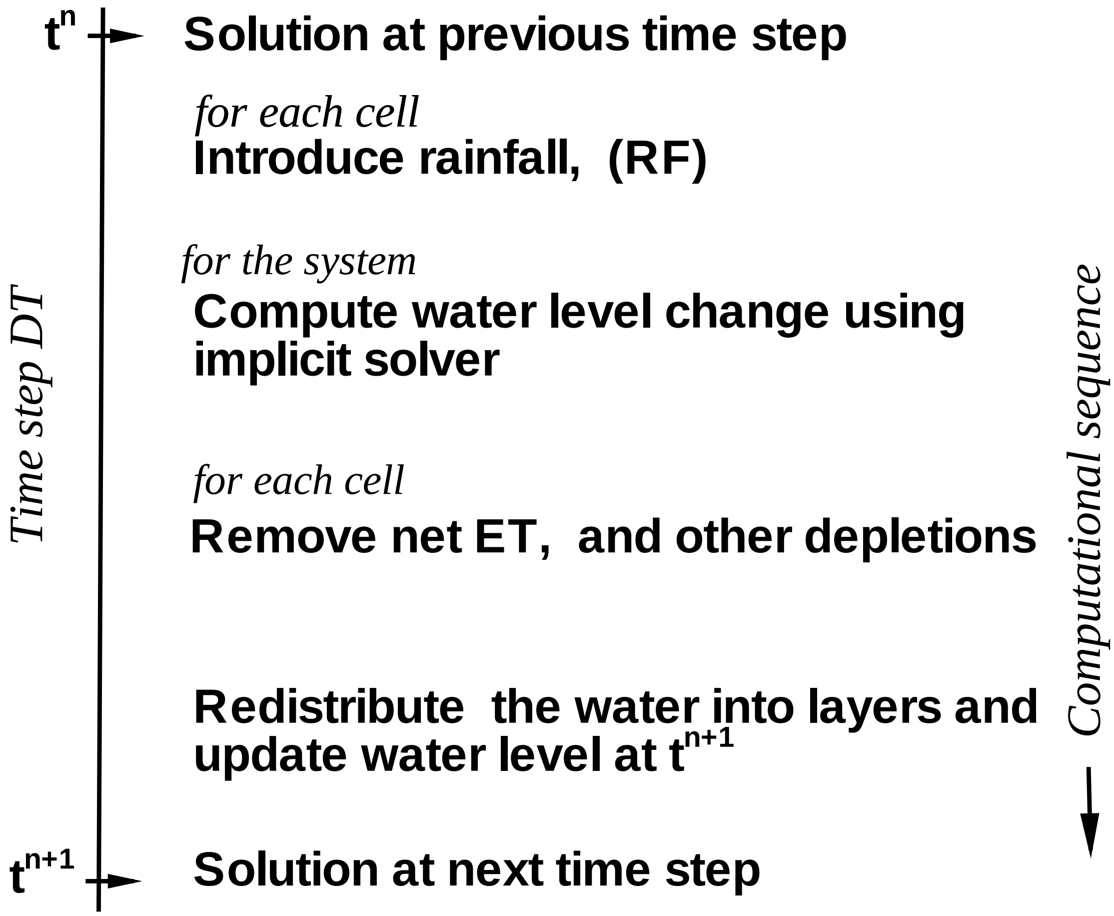

Figure 2.5: Sequence of steps used to interact horizontal and vertical solutions.

2.5 Numerical behavior of the vertical solution

The method used to carry out the vertical solution in the model is commonly referred to as a split method in which the complete equation

$$\frac{\partial H}{\partial t} = K\nabla^2 H + S(H) \tag{2.1}$$

is solved by using numerical operators designed to solve

$$\frac{\partial H}{\partial t} = K\nabla^2 H \tag{2.2}$$

and

$$\frac{\partial H}{\partial t} = S(H) \tag{2.3}$$

A first order splitting method is currently planned, in which

$$H^{n+1} = H^n + L_k(\Delta t).L_s(\Delta t) \tag{2.4}$$

in which, L_k, L_s are the operators for Eq. 2.2 and Eq. 2.3. L_k is the finite volume diffusion solver, and L_s is the vertical solution. Other methods of splitting (Strang, 1968) are not attempted.

The first order coupling works perfectly if S is not a function of H. However, if it is a strong function, stiffness will become a problem, and the numerical accuracy will begin to suffer. It can be shown that L_s requires a stability condition of the form

$$\Delta t \leq \frac{2}{\lambda} \tag{2.5}$$

in which, λ is the largest value of $\frac{\partial S(H)}{\partial H}$ in the model domain. S is defined as $RF - IN - ET$, and only IN and ET are functions of H. It is normally unlikely that these functions will create large values of λ to create inaccuracies or stiffnesses in the problem.

Chapter 3

Tests carried out with the overland flow module and the groundwater flow modules of the HSE

The accuracy of the two dimensional finite volume module of the HSE was tested by comparing its results with the results obtained using a number of other models. A number of test problems were used to carry out the experiment.

Test 1: Comparison of the HSE with the MODFLOW model

A test problem obtained from the text book by Wang (1982) was used in the comparison. In the test problem, a pumping well is located in the middle of a 4000m $\times$ 4000 m area. Tests were carried out for both confined and unconfined layer problems. When testing as a confined aquifer, the transmissivity was assumed as 300 m^2/day. When testing as an unconfined aquifer, a conductivity of 30 $m/s/day$ and a soil layer of 10 m were assumed, below which the soil was considered to be impervious. The storage coefficient used was 0.002. A uniform water level of 10 m was assumed as the initial condition. For the modflow simulation, $\Delta x = \Delta y = 100$ m were assumed. For the HSE simulation, a random triangular grid with 238 triangles and 135 nodes was used. In comparison, MODFLOW used 1600 squares to discretize the same area. Pumping rates used for the confined and unconfined aquifer tests were 2000 m^3/day and 1000 m^3/day respectively. For both tests, 1 day time steps were used in fully implicit modes ($\alpha = 1$). The simulation period used was 30 days. Following are the cell numbers and the reference coordinates of the monitoring sites.

Table 3.1: Test monitoring sites and their cell numbers.

Cell	Radial dist.	Co-ord.	Name
117	0.0 m	(0,0)	Well
84	600.8 m	(-596.8,-68.9)	Site 1
95	1010.8 m	(1010.8,-10.6)	Site 2
101	2006.2 m	(2006.1, -21.2)	Site 3

Figure 3.1 shows the triangular mesh used with the finite volume model. Figure 3.2 shows the time variation of the drawdowns at the well and the monitoring points during the 30 day period for the unconfined aquifer. Both finite volume and MODFLOW solutions shown are for the unconfined aquifer. Figure 3.3 shows the contour plot for drawdown near the unconfined aquifer, obtained using the MODFLOW model. Figure 3.4 shows the same drawdowns obtained using the finite volume model. Figure 3.5 shows the flow vectors at the end of 30 days.

The same tests were carried out with the confined aquifer too. Figure 3.6 shows the time variation of drawdowns at the same monitoring points. Figure 3.7 shows the drawdown contours obtained using the MODFLOW model. Figures 3.8 shows the same contours obtained using the finite volume model.

Test 2: Comparison of the HSE and axisymmetric solutions

The HSE overland flow solution for axisymmetric flow problem was compared with its axisymmetric model solution. The same comparison has been previously carried out to verify that the axisymmetric solution agrees with both NSM and SFWMM solutions. The test bed used has dimensions 160.9 km $\times$ 160.9 km (100 miles $\times$ 100 miles) and a flat bottom. The initial condition is

$$H = \left[0.4575 + 0.1525\cos(\frac{\pi r}{r_{max}})\right] m \quad \text{for} \quad r \leq r_{max} \tag{3.1}$$

$$H = 0.305\ m \quad \text{otherwise} \tag{3.2}$$

in which, $r =$ distance from the center; $r_{max} =$ 32188 m; $n_b = 1.0$. A no-flow boundary was assumed at the outer edge. The simulation time used was 12 days. A triangular mesh with 325 elements and 180 nodes was used in the finite volume solution. The time step used was 6 hrs, and $\alpha = 0.5$. The results were compared to the results of the axisymmetric diffusion flow model using $\Delta x =$ 80.5 m and time step 1 min.

Figure 3.9 shows the mesh used for the overland flow test. Figure 3.10 shows the time variation of water levels at the center (cell 107), and at radial distances 16522 m (cell 141) and 31945 m (cell 75) from the center. The figure shows that the finite volume solution and the axisymmetric solution are very close to each other. Figure 3.11 shows a contour plot of water levels at the end of the simulation. The figure shows that the circular patch of water remains circular during the simulation.

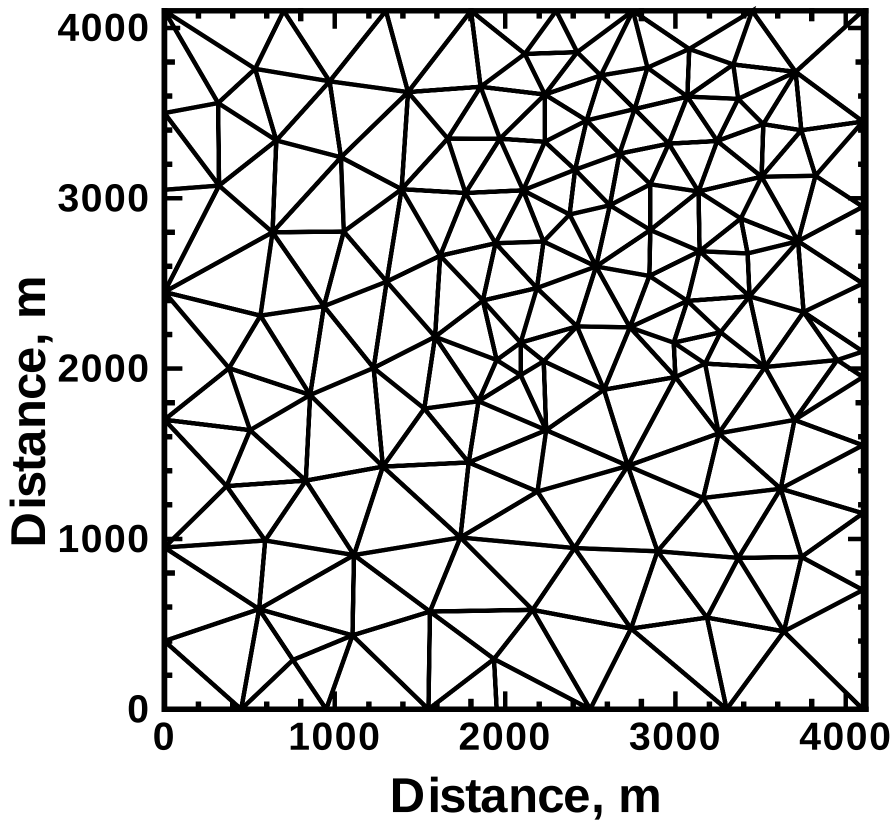

Fig. 3.1: The triangular mesh used to simulate the groundwater problem using the finite volume model.

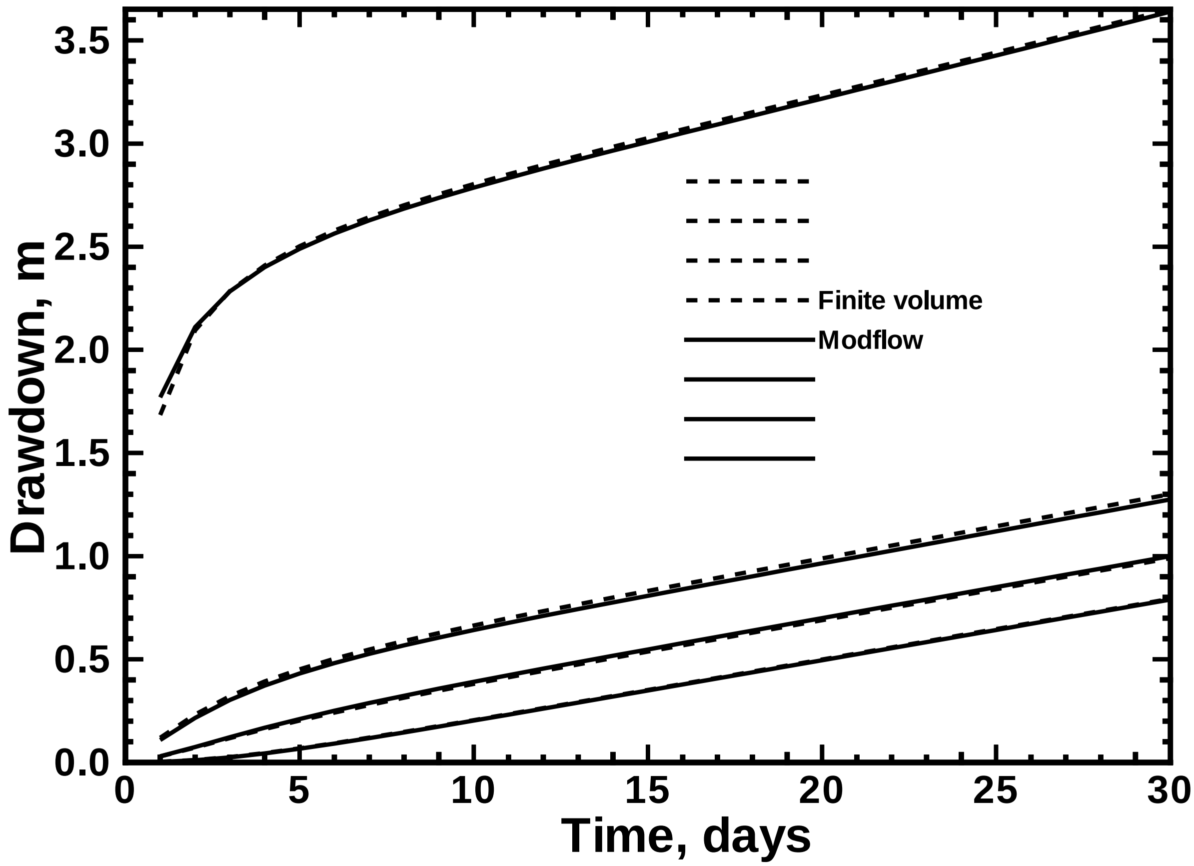

Fig. 3.2: Time variation of drawdown at the well and other points for the unconfined aquifer.

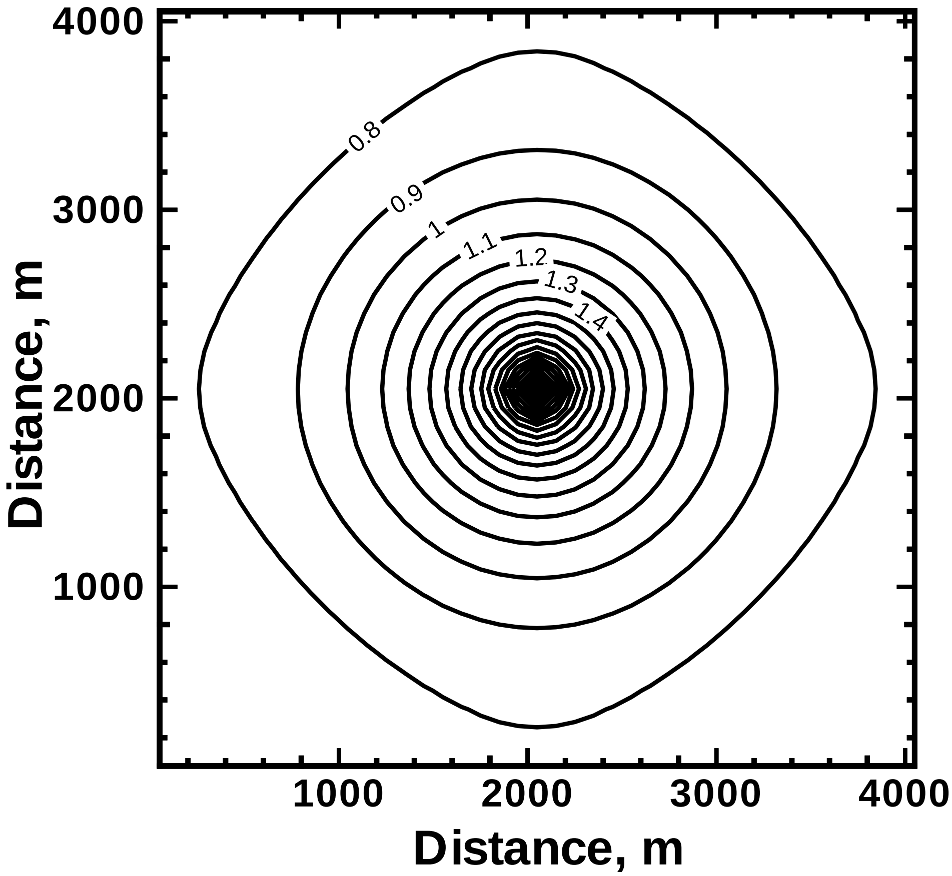

Fig 3.3: Drawdown contours obtained using MODFLOW for the unconfined aquifer.

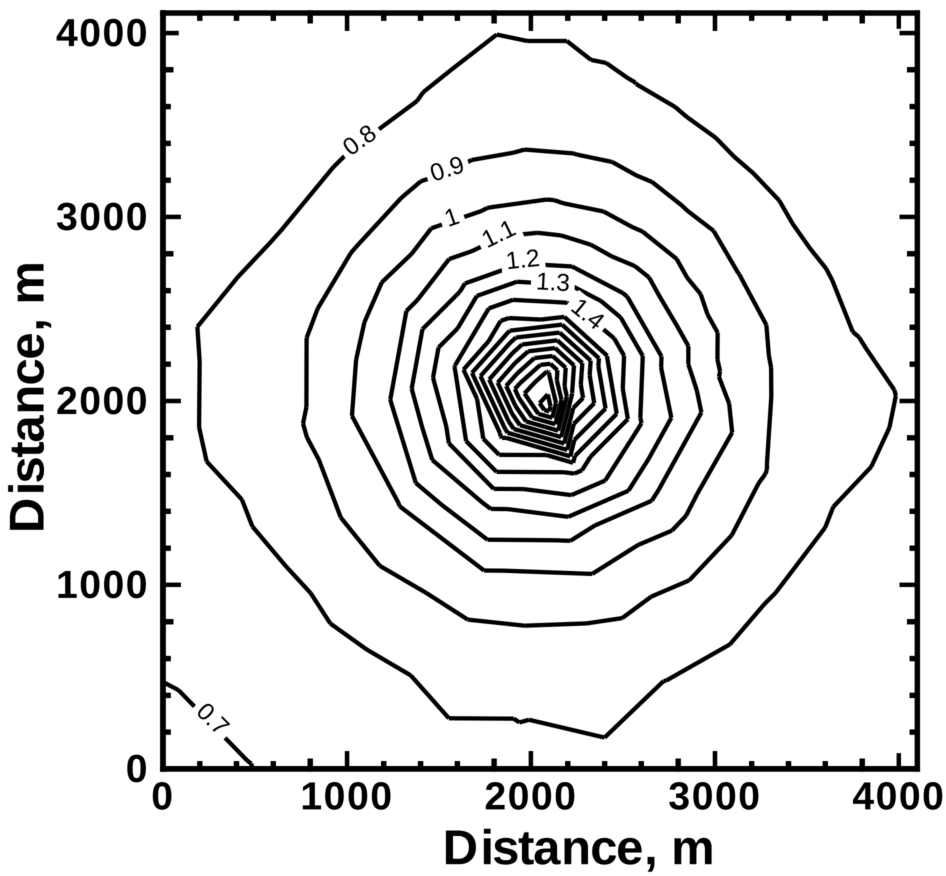

Fig 3.4: Drawdown contours for the unconfined aquifer obtained using the finite volume model.

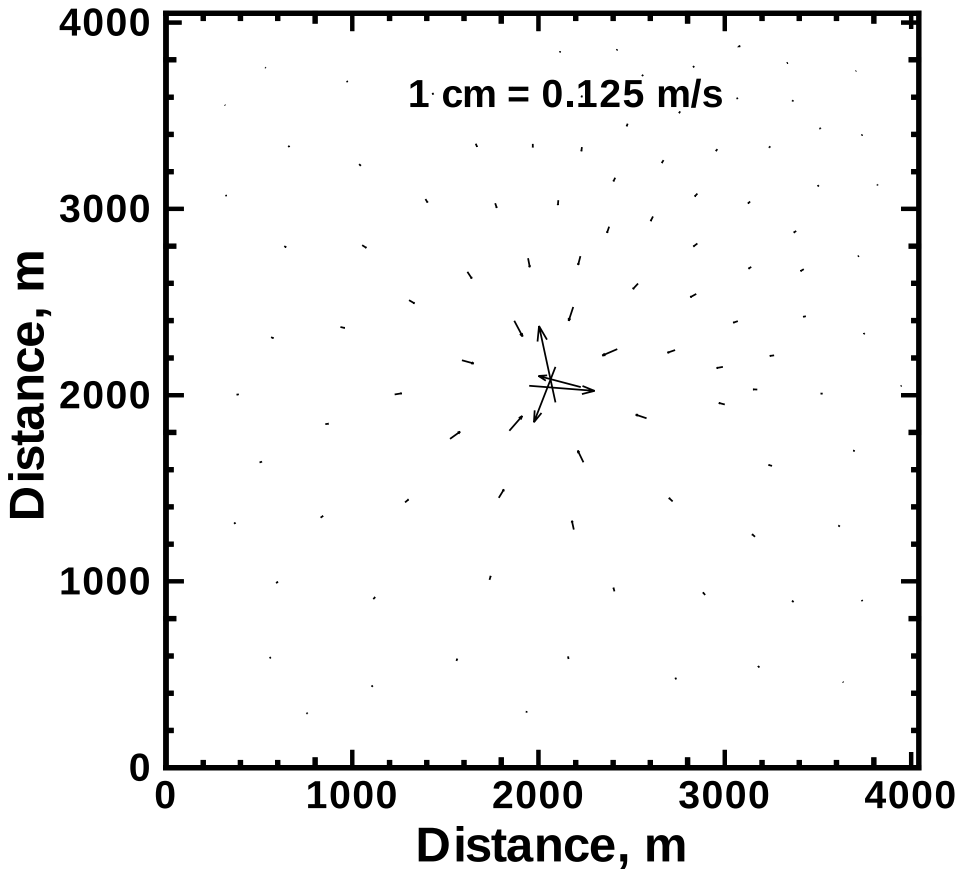

Fig 3.5: Flow vectors for the unconfined aquifer obtained using the finite volume model.

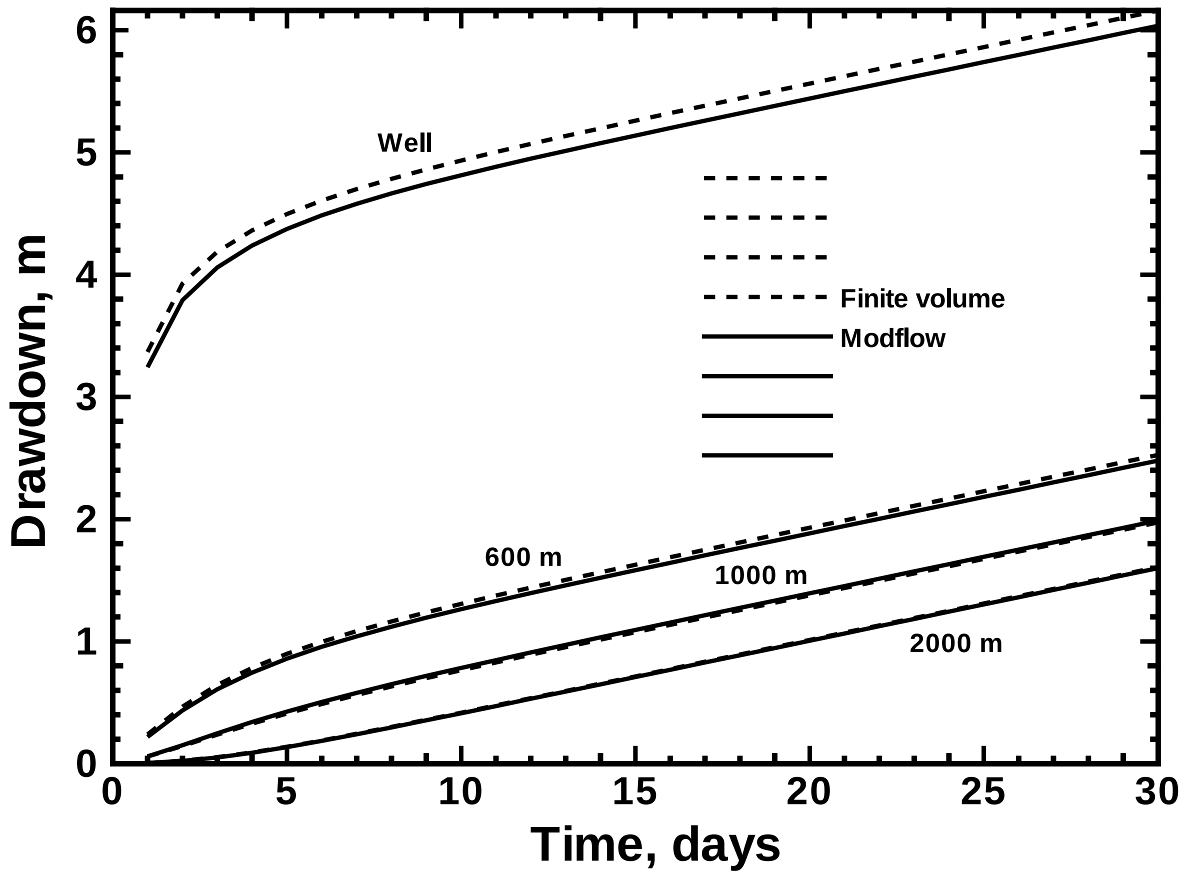

Fig 3.6: Time variation of drawdown at the well and other points for the confined aquifer.

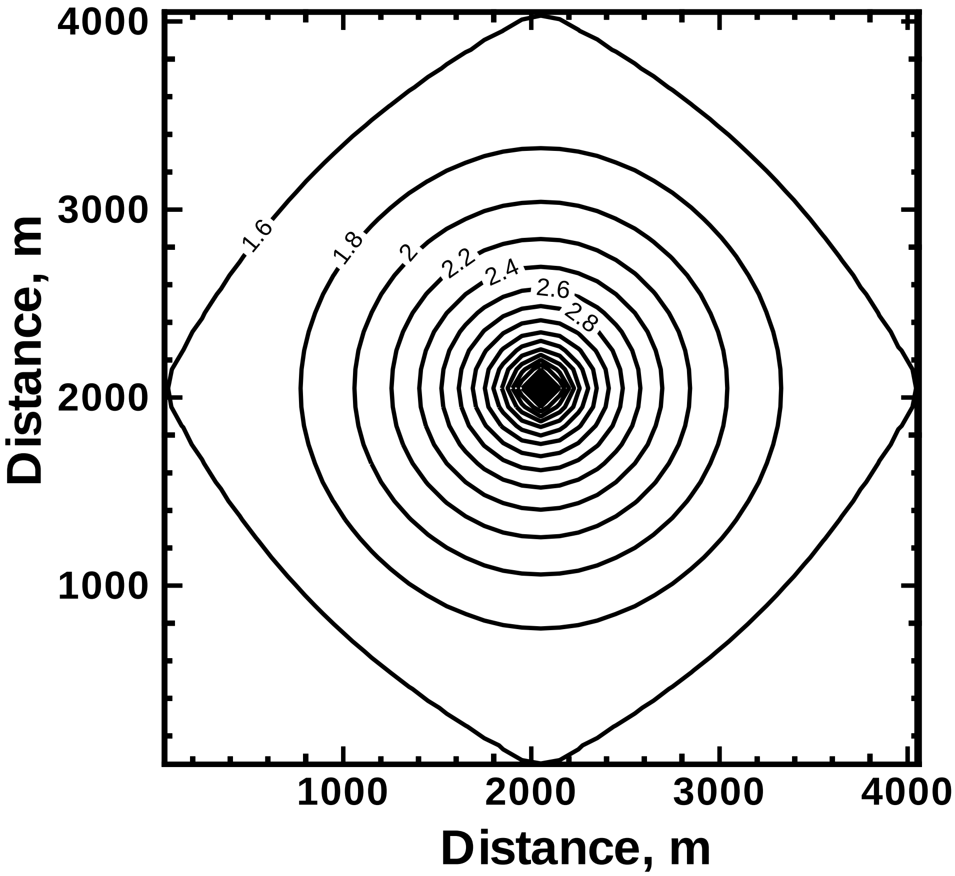

Fig 3.7: Drawdown contours for the confined aquifer obtained using the MODFLOW model.

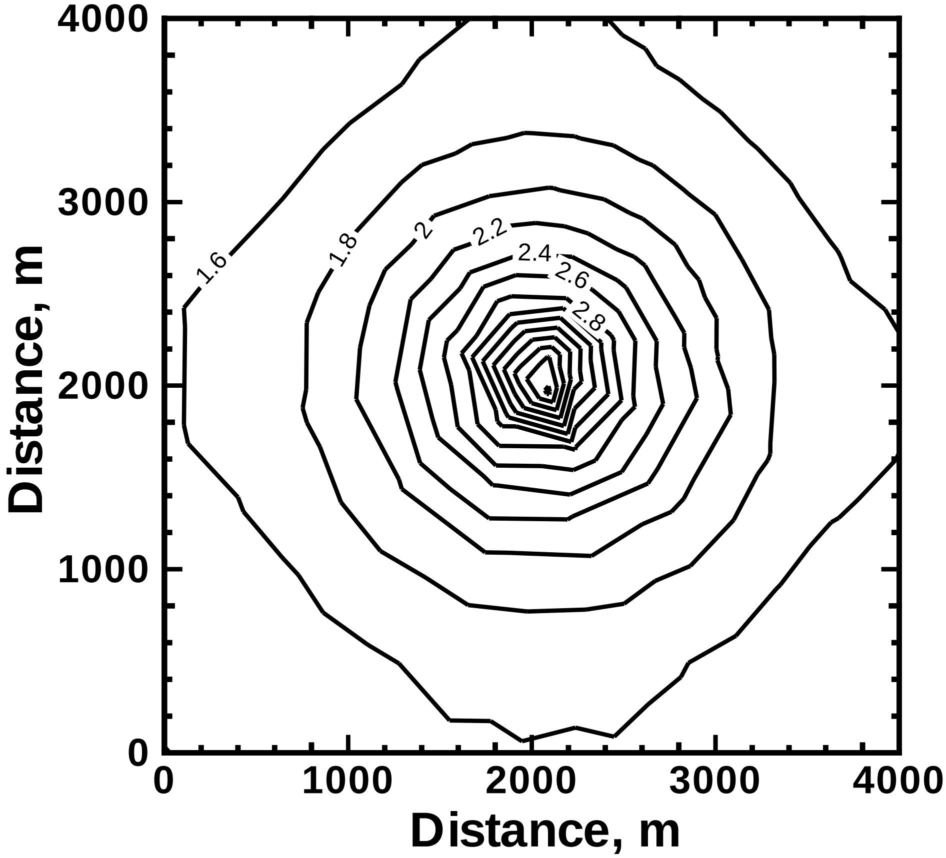

Fig 3.8: Drawdown contours for the confined aquifer obtained using the finite volume model.

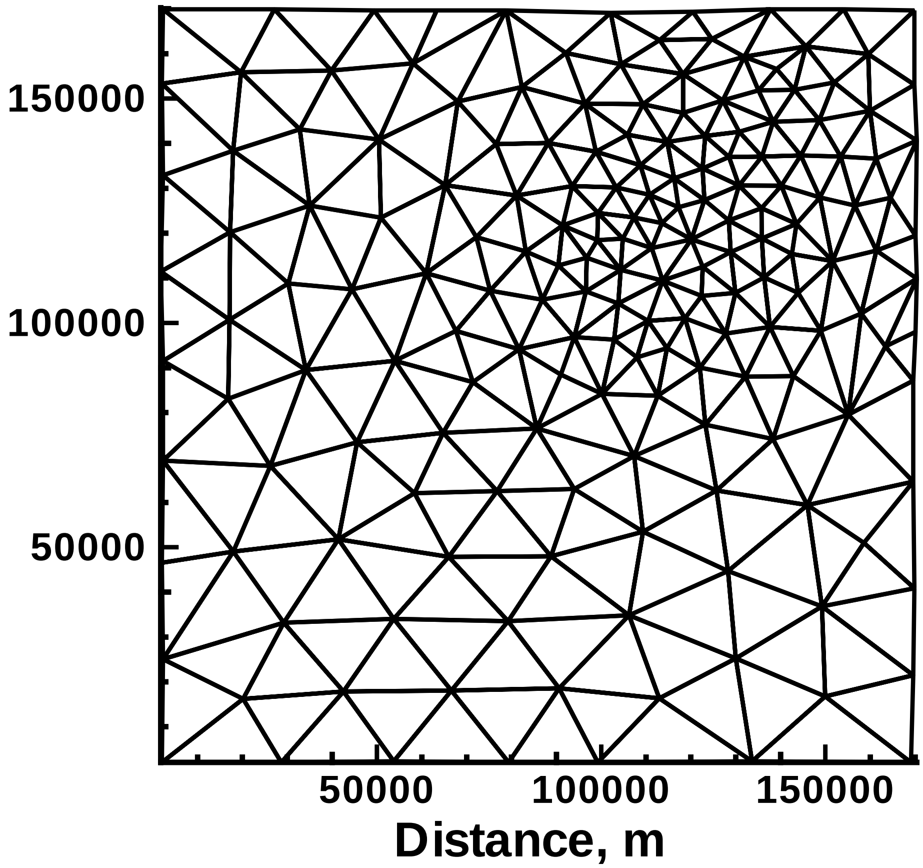

Fig 3.9: Triangular mesh used for the overland flow test.

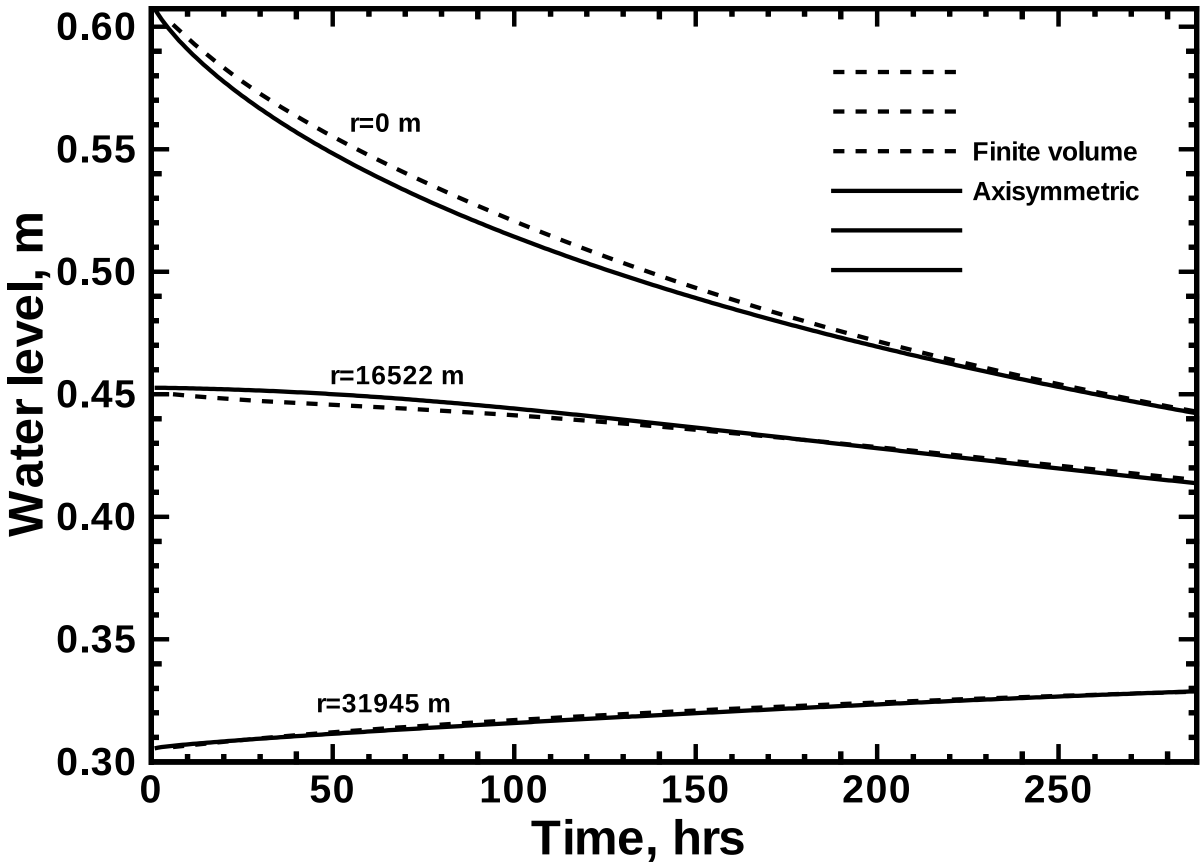

Fig 3.10: Variation of the water level in the overland flow solution with time.

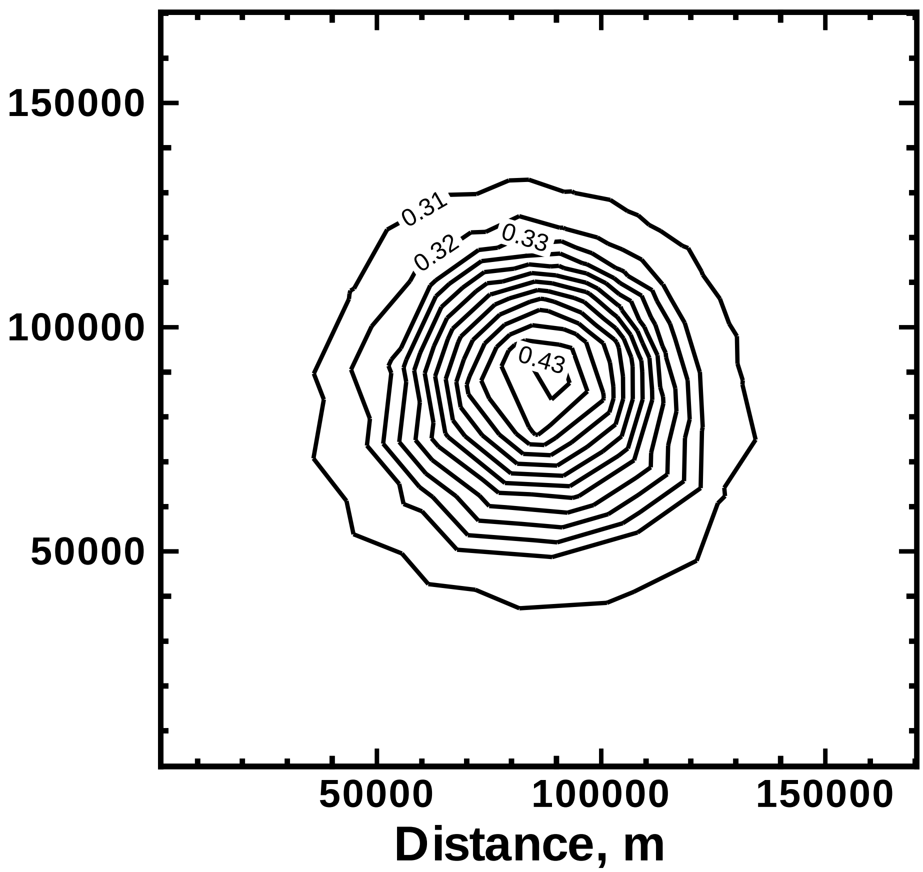

Fig 3.11: Contour plot of water levels in the overland flow solution.

Chapter 4

Numerical error analysis of the overland flow modules

The accuracy of the results obtained using the overland flow and the groundwater flow models depend on the spatial and temporal discretizations used. If the models are used to simulate flow features of a certain wave length, the resolution of the mesh should be sufficient to capture that flow feature. A description of the variation of the numerical error with the spatial and temporal resolutions is available in Lal (1996). In this paper, the experiments were conducted on models using rectangular grids. However, the results can be used in the current finite volume model as well.

In order to understand the behavior of the numerical error in the current cell centered finite volume model, triangular meshes of different sizes were used to simulate known flow patterns. The 160.9 km $\times$ 160.9 km square domain in Chapter 3 was used for the test because an accurate solution for it can be obtained using axisymmetric methods. Triangular meshes for the study were generated using the GMS package. Water level at the center of the circular patch was used to estimate the approximate numerical error in the solution. Table 4.1 shows a summary of test results, including the number of cells, nodes, and run times.

The results in Table 4.1 can also be expressed in terms of the non-dimensional pa-

Table 4.1: Solutions of the test problem obtained with various discretizations. * indicate the test case shown in the previous chapter whose grid is not homogeneous.

Test	No. elem.	No. nodes	CPU (s)	No. iter.	Δx (m)	Δt (s)	h_{end} (m)	π/ϕ	β	ϵ %
1	116	69	2.4	18	14939	51840	0.44877	2.15	0.0164	1.09
2	116	69	8.8	12	14939	10368	0.44840	2.15	0.0033	1.03
3	116	69	16.4	11	14939	5184	0.44840	2.15	0.0016	1.02
4*	238	135	10.3	1	10429	5184	0.43921			0.50
5*	238	135	15.7	1	10429	10368	0.43908			0.50
6*	238	135	27.7	1	10429	5184	0.43901		0.50	0.49
7	376	209	6.0	40	8298	207360	0.44500	3.88	0.2121	0.48
8	376	209	25.1	19	8298	20736	0.44456	3.88	0.0212	0.40
9	376	209	43.6	17	8298	10368	0.44444	3.88	0.0106	0.38
10	376	209	78.8	13	8298	5184	0.44438	3.88	0.0053	0.37
11	1536	809	60.1	104	4105	518400	0.45404	7.84	2.1660	1.96
12	1536	809	75.3	78	4105	207360	0.44494	7.84	0.8660	0.48
13	1536	809	98.3	67	4105	103680	0.44501	7.84	0.4332	0.48
14	1536	809	258.0	35	4105	20736	0.44388	7.84	0.0866	0.29
15	1536	809	436.0	27	4105	10368	0.44374	7.84	0.0433	0.27

rameters derived in the error analysis by Lal (1996). Since triangular cells are used in the problem instead of square cells, Δx in the table was obtained as $\sqrt{A_c}$ in which, A_c is the average area of a triangular cell. ϕ is obtained as $k\Delta x$ in which, k is the wave number of the water surface profile simulated in the model. k is computed as $2\pi/\lambda$. The expression π/ϕ gives the average number of spatial divisions within half the length of a sine wave. β is computed as

$$\beta = \frac{h^{\frac{5}{3}}}{n_b\sqrt{S_s}}\frac{\Delta t}{\Delta x^2} \tag{4.1}$$

Table 4.1 shows that the behavior of the error is similar to the behavior of the error observed in the paper (Lal, 1996). An estimate of the error obtained by expressing the error at the peak of the sine wave as a percentage of the maximum depth increases when (π/ϕ) is decreased, and β is increased. Figures 4.1, 4.2 and 4.3 show the contour plots of the water levels at various spatial spatial resolutions (π/ϕ) of 2.15, 3.88 and 7.84. Results show that with the higher resolution (π/ϕ) of 7.85, the estimated error can be less than 1%.

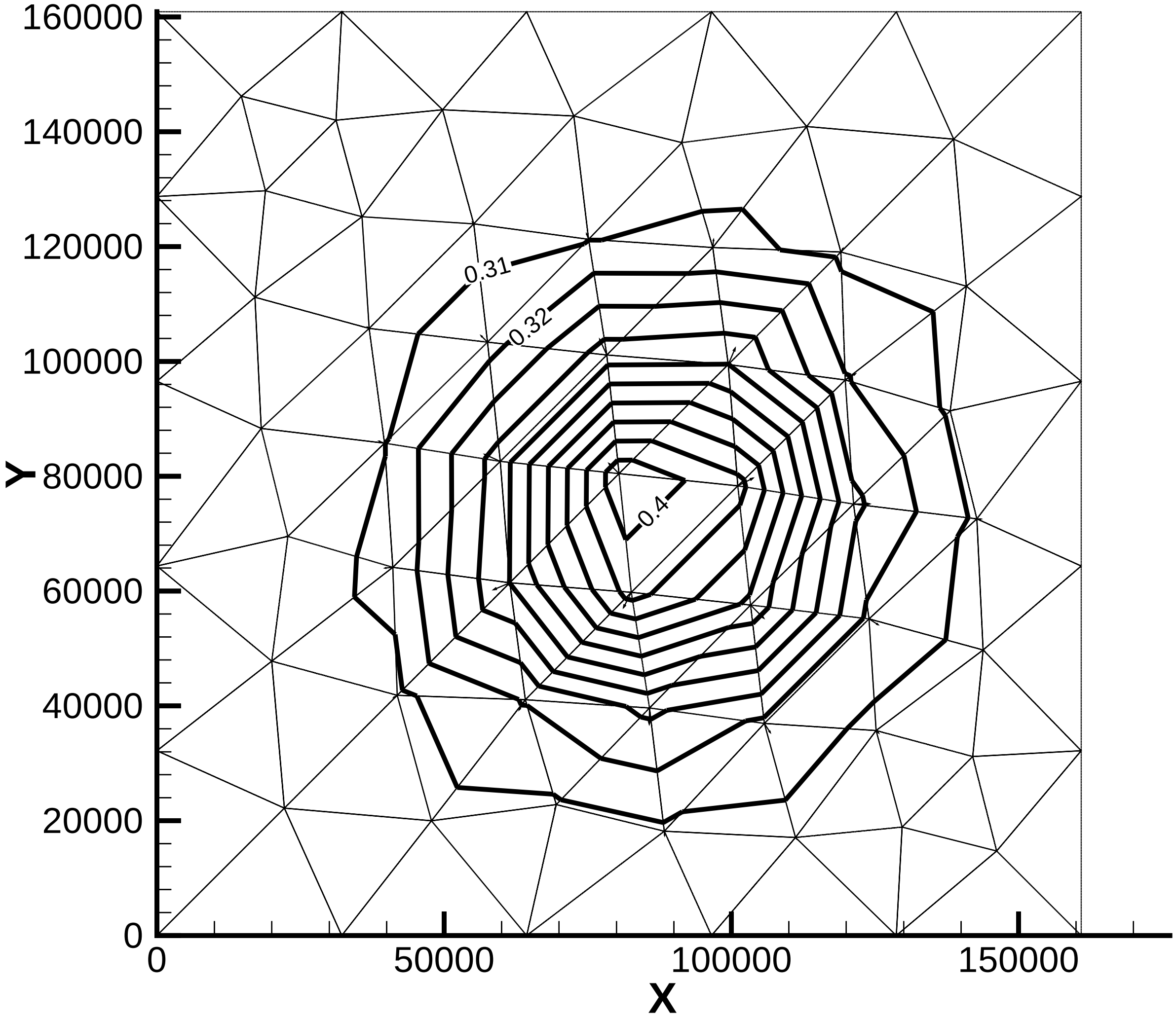

Fig 4.1: Mesh and the contour plot for the discretization with 116 cells

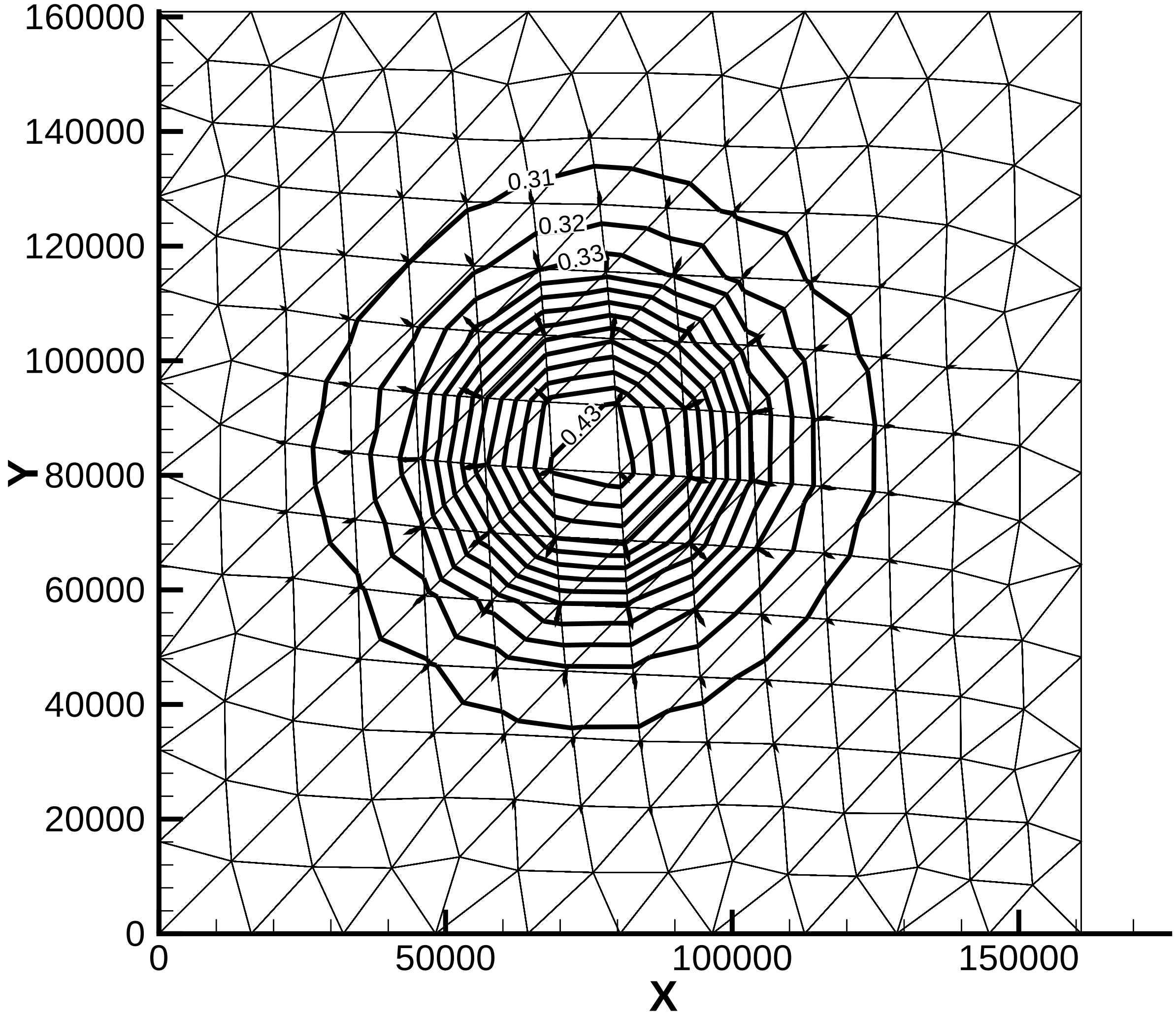

Fig 4.2: Mesh and the contour plot for the discretization with 376 cells

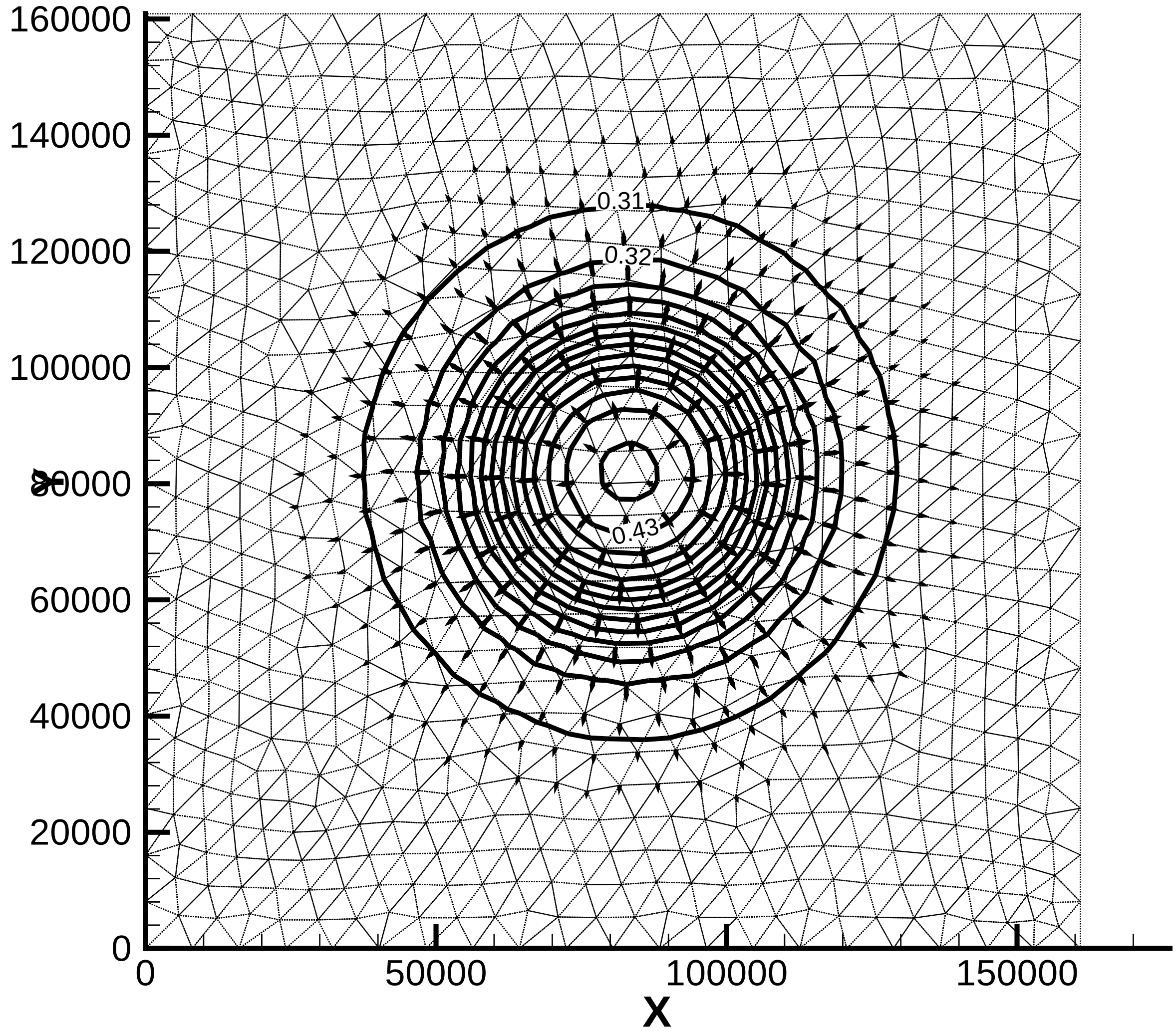

Fig 4.3: Mesh and the contour plot for the discretization with 1536 cells

Chapter 5

One dimensional canal network model

The one dimensional canal network model of the HSE is capable of simulating canal or river flow in connected or disconnected canal or river systems. The one dimensional flow is approximated using diffusion flow equations, assuming that the inertia terms can be neglected. The importance of the inertia terms in canal systems will be studied separately to understand the limitations of the diffusion flow assumption. The HSE network model is capable of simulating flow through structures and junctions, while considering the effects of different head and discharge boundary conditions. In a coupled system consisting of 2-D overland flow and 1-D canal flow, the canal system is laid over the 2-D system, and the interaction terms are computed based on the water levels at different overlapping elements.

5.1 Discretization of the canal system

Solution of the St. Venant equations, or its simplified form, the diffusion flow equations, requires discretization of the canal system. An ideal discretization should have uniformly sized and shaped canal sections with high resolutions at locations where the user needs a refined solution. The optimum level of the spatial discretization can be decided based on the work by Lal (1996).

Figure 1 shows an example of an discretization of a canal system using the current finite volume method. As the first step, the river or the canal is divided into a number of sections or segments of somewhat equal length. Uniform segments may help to improve the condition of the matrix, and therefore the accuracy of the solution and the speed of computations. When the flow enters the model, it should enters at the end of a boundary segment. In the figure, segments begin at cross section c1 and end at cross sections c9 and c13. The first segment marked with italic 1 is bounded by cross section c1 and c2, etc. These last two sections are associated with boundary conditions such as uniform flow or river flow, etc.. Each of the river segment has cross section information at either end. For segment 1, the river cross sections are at c1 and c2. A node or a joint is placed at the ends of all river sections. Nodes are necessary at structures, joints, and the beginnings and ends of canal segments. The main purpose of the nodes is to describe the configuration of canal networks. A single node 4 is placed near river cross sections c4, c5 and c6 in the figure asuming that these sections are very close.

The average properties of a segment should be used when describing a cross section. Such average properties can be obtained by using a single section halfway in the segment, or averaging the properties at its end sections. In the case of segment 1, sections c1 and c2 at its ends can be used to obtain average properties. Lakes are represented by river sections with extremely small resistances, and finite plan areas. This makes it easy to accomodate lakes easily in the network model.

Only two canal segments are attached to a structure, one at the upstream end, and one at the downstream end. This helps to simplify the model formulation. If there are structures connected to more than two segments, a small canal segment has to be introduced to create a new junction, and the short segment should branch off the main canal. The short length would provide a small resistance, and would have the same effect as having two canal segments are connected to the same structure.

Figure 5.2 shows an example of a canal discretization in elevation. The dotted line shows the ground profile. The figure shows how the uniform segments with average properties would appear in a conceptual diagram. Since the water level and the ground elevation are assumed constant within a conceptual cell, the figure helps to explain the positions of the canal bed and water surface in the field and in the model. In the figure, the water surface and ground elevations at the interior canal segments are computed at the mid points of river segments. As a result, flow between two internal segments takes place approximately between the two centroids of the segments. The elevations at the centroids provide the actual elevations found in the field. However, at the end segments such as segment 1 at which the boundary conditions are applied, water and bed elevations are computed at the for end of the segment instead of the centroid, so that the full length of the canal is considered when accounting for friction.

When using the finite volume method to simulate the canal systems, the discretized canal segments are considered as individual control volumes having the required average cross sectional properties, and correct open water areas. Segments are connected to each other at the cell walls at each end. An imaginary cell wall is assumed to exist between every pair ov cells connected at a node. In other words, a flow pair is assumed to exist between every pair of cells connected. The friction relationships between these pairs of cells form the basis of all flow computations in the network. The relationship takes the form $\Delta Q = K \Delta H$, in which, $\Delta Q, K$, and ΔH are discharge, resistance and head loss between the segments. In In the case of structures, the friction is computed uding the structure equations. In the case of canal segments, Manning's equation is used to compute the friction between the centroids of the neighboring cells.

The friction relationship for two interior canal segments i and j is

$$Q = K_{i,j}(H_i - H_j) \tag{5.1}$$

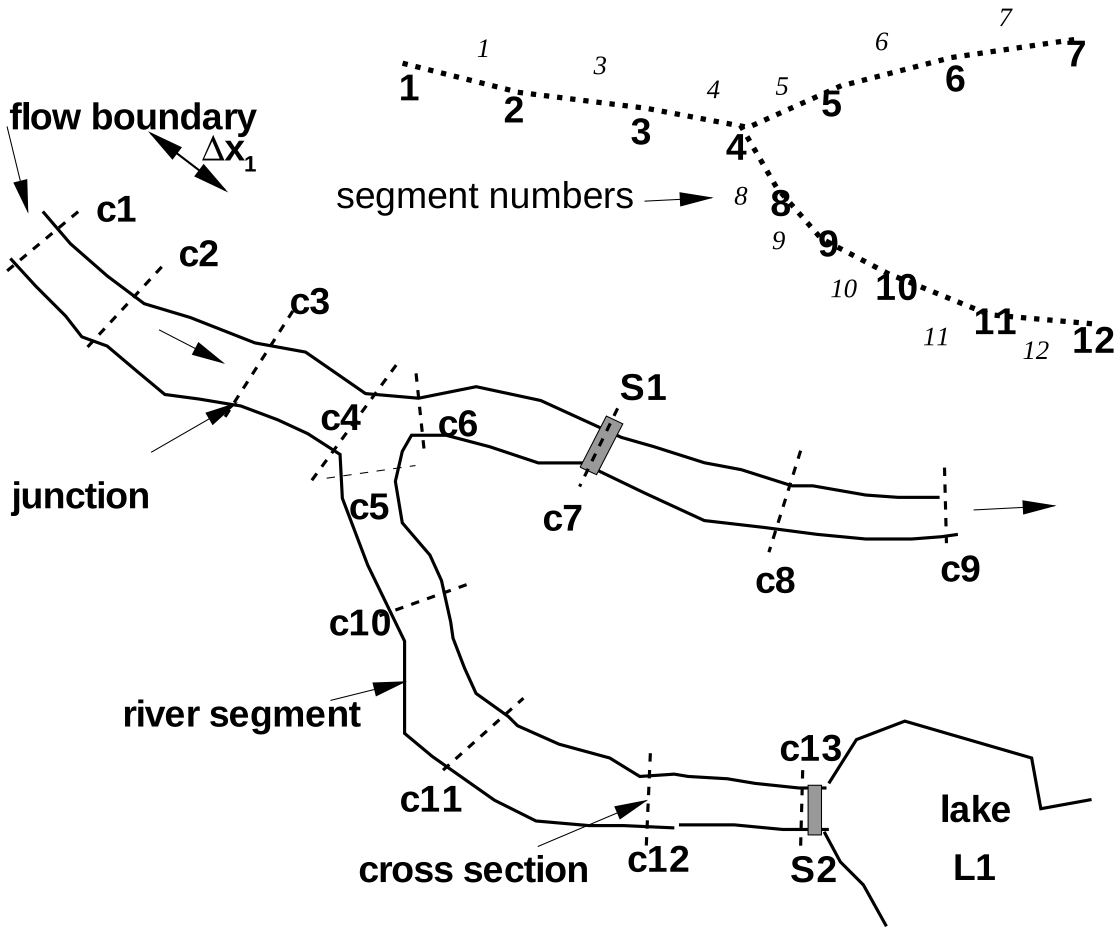

Fig 5.1: Figure showing an example of a canal discretization.

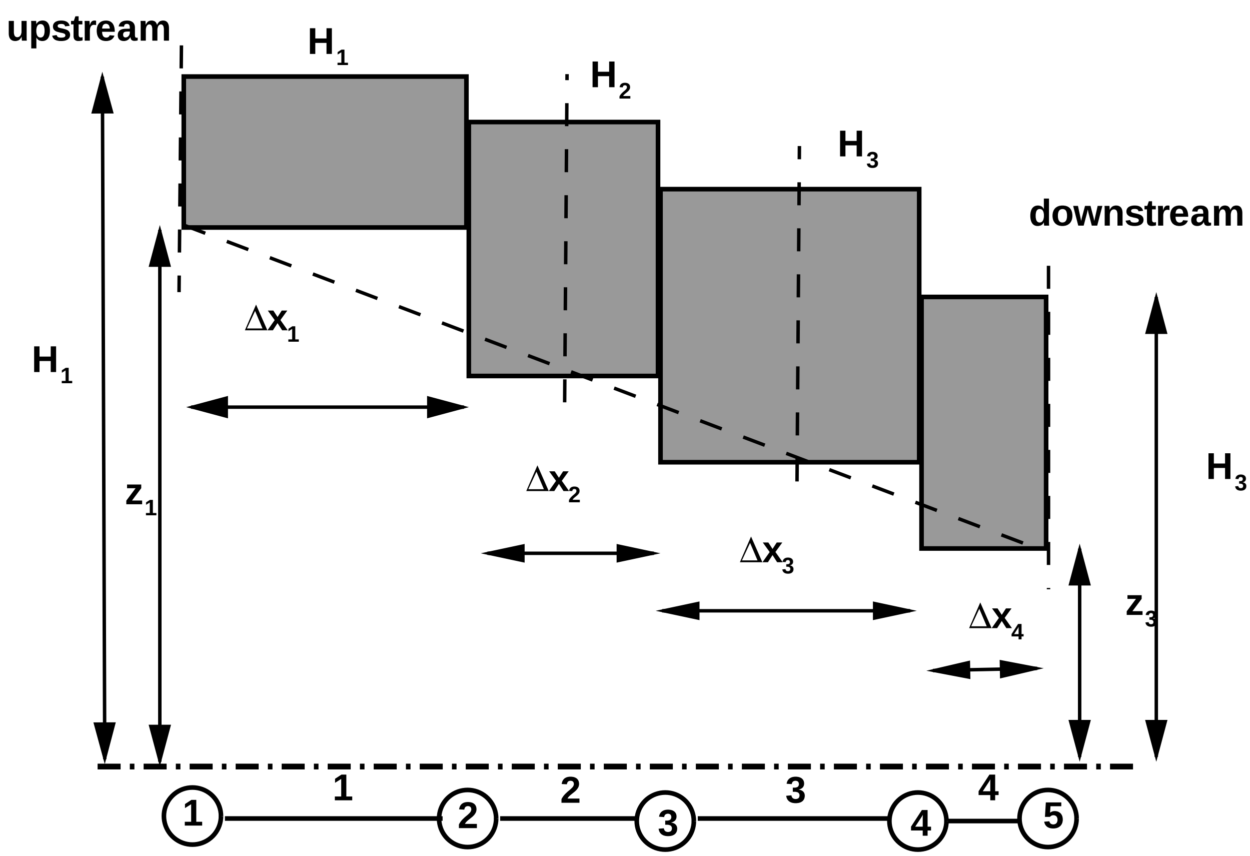

Fig 5.2: Figure showing an example of a longitudinal section of a canal.

in which

$$\frac{1}{K_{i,j}} = \sqrt{0.5(\frac{\Delta x_i}{K_i^2} + \frac{\Delta x_j}{K_j^2})\Delta H} \tag{5.2}$$

K_i, K_j are the average K values for the segments ($K = AR^{\frac{2}{3}}/(n\sqrt{S}$ when Manning's equation is used.) In the case of canal segment 1 in Fig. 5.1, the effective length Δx_1 has to be doubled if Eq 5.2 is to be used without modification because it is an end segment. In the case of structures having an equation similar to Eq. 5.1 in which K_s is placed intead of $K_{i,j}$, the added resistance to the flow due to the canal lenghts adjacent to it would result in an effective resistance K_s' given by

$$\frac{1}{K_s'} = \frac{1}{K_s} + \frac{0.5\Delta x_i}{K_i} + \frac{0.5\Delta x_j}{K_j} \tag{5.3}$$

If the structure has more resistance to flow compared to the adjacent river segments ($K_i, K_j >> K_s$), it is possible to assume $K_s' = K_s$. If the structure is adjacent to a lake as in S2 of Fig 5.1, K_{12} becomes extremely large, and would disappear from Eq. 5.3.

There are many similarities between the finite volume building blocks or cells used in the two dimensional model and the one dimensional network model. The similarities can be explained using the following transformations of components which in general lead to the reduction of the dimensionality by one.

$$\begin{aligned} \text{polygons in 2-D} &\rightarrow \text{Canal segments (line elements) in 1-D} \\ \text{flow walls in 2-D} &\rightarrow \text{flow walls (at nodes) in 1-D} \\ \text{nodes in 2-D} &\rightarrow \text{none} \end{aligned} \tag{5.4}$$

The numerical formulation of the finite volume method is also guided by conservative principles as well. In the formulation, canal segments or 1-D elements are used to collect water, and canal walls are used to control flow across canal segments. The 1-D wall types include structure walls, 1-D flow walls, and no-flow walls.

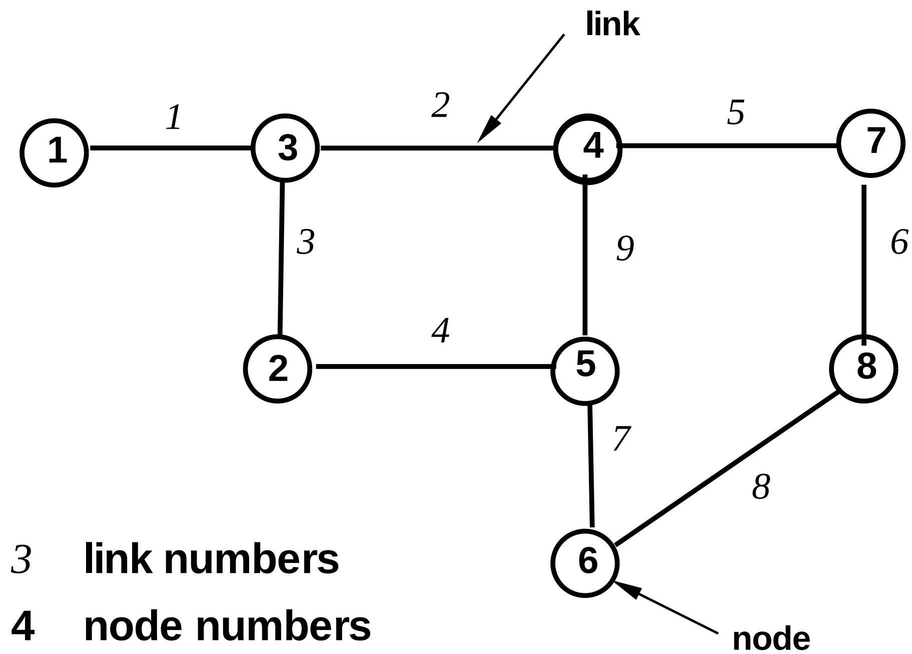

Fig 5.1: Figure showing the discretization of a canal system.

Table 5.1: Representation of the 1-D canal network using node and segment numbers.

Node	No. of seg.	seg1	seg2	seg3	seg4
1	1	1	0	0	0
2	2	3	4	0	0
3	3	1	2	3	0
4	3	2	5	9	0
5	3	4	9	7	0
6	2	7	8	0	0
7	2	5	6	0	0
8	2	6	8	0	0

5.1.1 Input data describing the network configuration

After discretization, information about the network is described using a geometry data file. Table 5.1 shows part of the data file describing the canal network discretization in Fig 5.3. The information is arranged against the node numbers. Column 2 of the table shows the number of segments attached to the nodes. Columns 3 and above show the segment numbers which are attached to the nodes. In Table 5.1, node 3 for example has three segments attached, which are marked as 1, 2 and 3. Appendix B shows the actual input data file used with Fig. 5.1

5.1.2 Internal representation of the canal configuration

The internal representation of model configuration data is different from the user supplied input data explained in Table 5.1. The internal data is created within the code by the pre-processor. It is in a form that is most efficient for the code to use. When structures are imposed, the default wall types assigned by the pre-processor are replaced by the imposed structure types. The default types are the 1-D flow type at internal walls, and the no-flow type at open ends. The 1-D flow type is computed based on diffusion flow equations. During pre-processing, all the 1-D cell walls in a network are numbered and

Table 5.2: The internal representation of a canal configuration

1-D Cell wall no.	u/s segment	d/s segment	Type	Sequence No.
1	1	3	1	
2	1	2	1	
3	2	5	1	
4	2	9	1	
5	4	9	1	
6	4	7	1	
7	5	6	1	
8	6	8	4	1
9	7	8	5	1

Some segment types used in the FORTRAN code:

0 = no-flow

1 = canal flow based on the Manning's equation

4 = structure of a selected type.

5 = structure of a selected type. ....

their wall types are assigned to default types. The segments connected by the 1-D walls are also determined. Table 5.2 show an example of this internal configuration. In this configuration, the wall 8 of type 4 connects segments 6 and 8. The boundary condition information file re-assigns all default wall types in a canal network.

5.1.3 Boundary condition data

Head and flow boundary conditions are the most commonly used boundary conditions in diffusion flow models. Flow control structures classified as internal boundary conditions are placed at walls separating the segments. Head and flow boundary conditions are specified for 1-D segments. They will force the water levels to take values specified by input time series. The sample input data file in Table 5.3 specifies the segment numbers, and the time series sequence numbers of the boundary condition data file. In the case of flow boundary conditions, specific flow rates are related to the segments specified by the boundary condition data file. An input data set for flow boundary conditions is very similar to the data set shown in Table 5.3.

Table 5.3: Example of a head boundary condition.

Segment No.	Time series sequence no.
3	1
5	2
6	2
....	

When structures are specified at 1-D walls, the input data file should contain the numbers of the two connected segments, and the new wall type. Table 5.4 shows an example of a boundary condition file used to describe Figure 5.1. The table shows structures of type 4 and 5 between segment sets (6,8), and (7,8).

Table 5.4: Example of a structure type internal boundary condition.

Seg 1	Seg 2	wall type	Sequence
6	8	4	1
7	8	5	1
....			

5.1.4 Description of canal cross sections in the model

In the current finite volume model, canals are considered to be made of canal segments of uniform cross section. Water levels and other parameters are also assumed to be constant within a segment. Two of the most important canal characteristics used in models are the cross sectional area and the width. Both of these change with the water level, and have to be computed at every time step. There are many methods available to provide this information to a model, even if there are only a few ways they can be used. In most models, they are used only to compute the area A, the width B, and the partial derivatives with respect to H. In order to provide the user the capability to use a variety of data formats in the current model, provisions are made for the user to write a simple routine to read the cross section data, and obtain expressions for A, B, and R.

```
      SUBROUTINE CSECT (H,Z,AREA,WIDTH,HYD_RAD)

.......obtain the information you need ....

.... write your routine ....

        AREA = write your expression
        WIDTH = write your expression
        HYD_RAD = write your expression
      RETURN
      END
```

In the sample data file shown in Appendix B, the expressions used to compute the area and the width are

$$B = B_0 + B_1(H - H_0) \tag{5.5}$$

$$A = A_0 + A_1(H - H_0) + A_2(H - H_0)^2 \tag{5.6}$$

$$\tag{5.7}$$

in which, H_0 is a datum; B_0, B_1, A_0, A_1, A_2 are coefficients that can be determined using regression or analytical methods. $B_1 = A_2 = 0$ was assumed in the example, meaning that the embankments are vertical.

5.1.5 Water level and flow boundary conditions

Water level boundary conditions can be assigned by listing the segment number and the time series data file number in the boundary condition data file. In the case of water level boundary conditions, the specified segment will set the head specified in the head time series data file. In the case of a flow boundary condition, the specified flow will be added to the segment.

Chapter 6

Tests carried out with the 1-D canal network model

The accuracy of the current 1-D weighted implicit diffusion model for canal networks was tested using a number of methods. A 1-D problem with a known solution was used for the first test. The known solution was compared with the solution obtained using a full equation explicit model and the current diffusion model. Both the test problem and the explicit model were obtained from the text by Viessmann, et al., (1977). Some other tests were also carried out to compare the solutions of the canal network model and other models.

6.1 Test 1 for a single canal stretch:

The problem used for the first test was obtained from the text written by Viessman, et al., (1977). The problem is stated below.

A 20 ft wide rectangular channel 2 mi long having uniform flow of 6 ft depth is subjectedto an upstream increase in flow of 2000 cfs in a period of 20 min. This flow then decreases uniformly to the initial depth flow depth in an aditional period of 40 min. The channel has a bottom slope of 0.0015, an an estimated Manning's n of 0.02. Calculate the explicit solution of hydraulic routing for this situation.

The explicit method and the method of characteristics are used in the text to solve the above problem. The time step used in the explicit method was 2 s. The time step of the characteristic method was variable, but of the same order. The time step used with the current weighted implicit solver was 6 s. The solver was found to be stable unconditionally, and the results were accurate even with 120 s time steps. The canal was discretized into 40 sections for both methods. Figures 6.1 and 6.2 shows the variations of water levels and discharges at the upstream, midstream and the downstream points when using both methods. Even if the diffusion model does not consider the inertia terms, figure shows that the results of both models agree with each other.

The data file for the example is not included. But a smaller set of data from a similar aplication with 4 spatial discretizations instead of 40 discretizations is shown in Appendix C.

6.2 Test 2 for a branch channel

Flow split at a canal joint duting the passing of a flood peak was studied, and compared with the results obtained for UNET. The example consists of a 1 mile channed splitting into a 2 mile channel and a 3 mile channel. The properties of the three straight channels are shown in Fig. 6.3. The bottom elebations at the ends are also shown. Manning's roughness was assumed to be 0.02 for all channels. The flow hydrograph at the inflow begins as steady flow of 589.7 m^3/s, and increases linearly from 589.7 m^3/s to 1415.8 m^3/s in 20 minutes and decreases linearly to 589.7 m^3/s in 40 more minutes. The downtream ends have head boundary conditions, with water levels specified as 2.017 m and 1.213 m at sections 2 and 3.

Figures 6.4 and 6.5 show a comparison of the diffusion flow model results (HSE) with UNET results. The input data files for UNET are shown in Appendix E. The space and time steps used in the diffusion flow model were 6 minutes and 804 m. In the UNET model, they were 2 mins and and 804 m respectively.

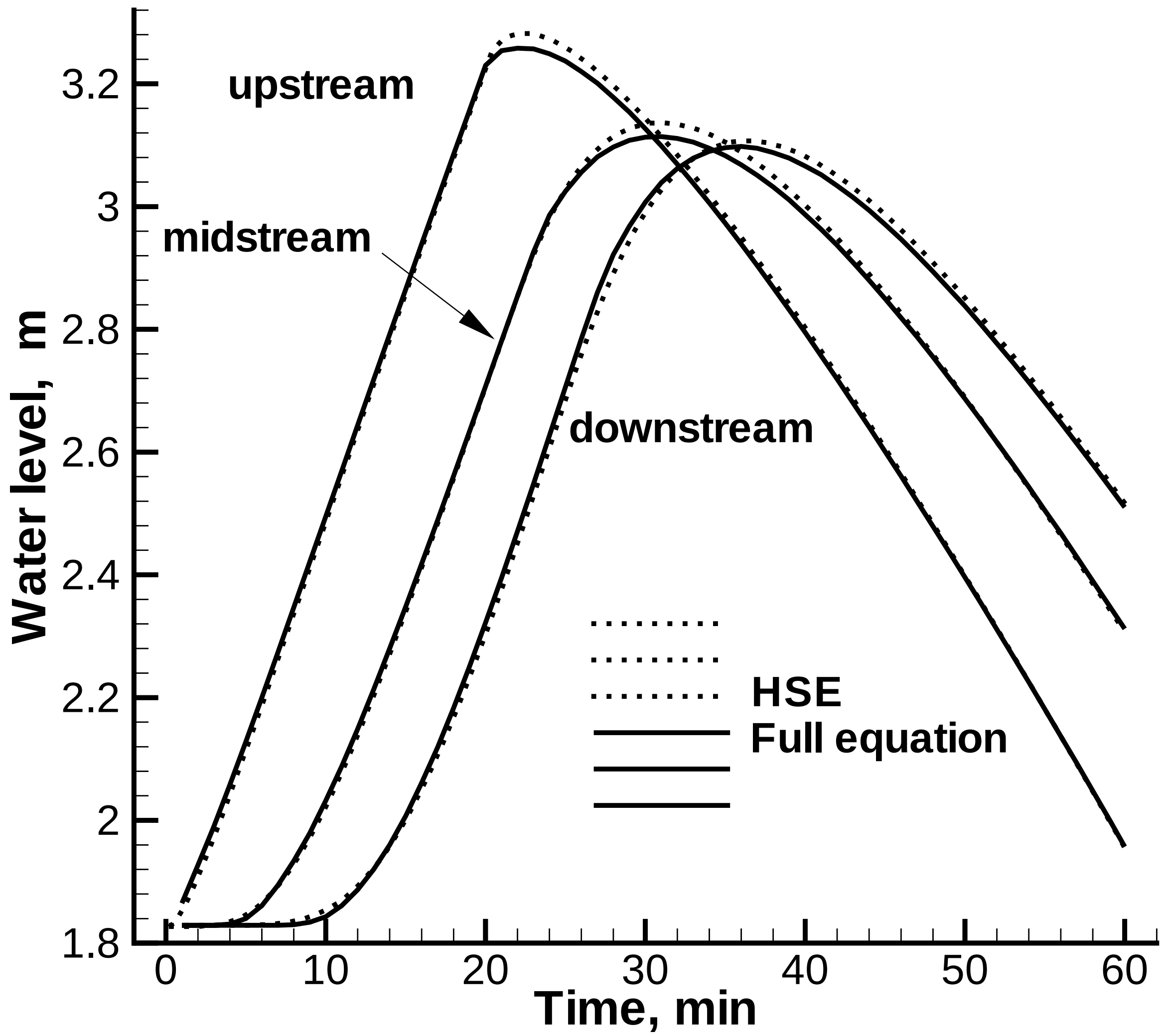

Fig 6.1: Variation of the water levels with time.

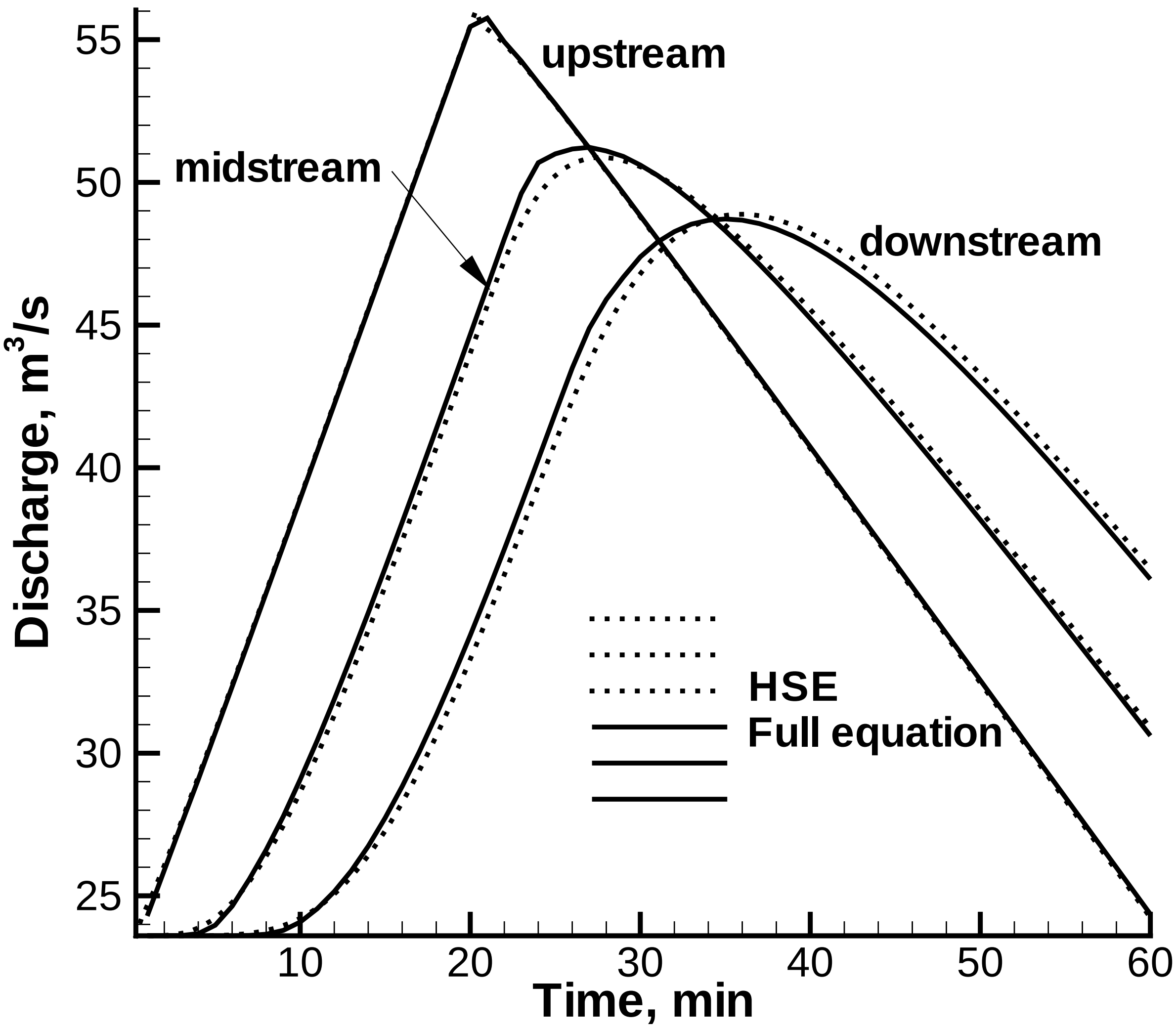

Fig 6.2: Variation of the discharge with time.

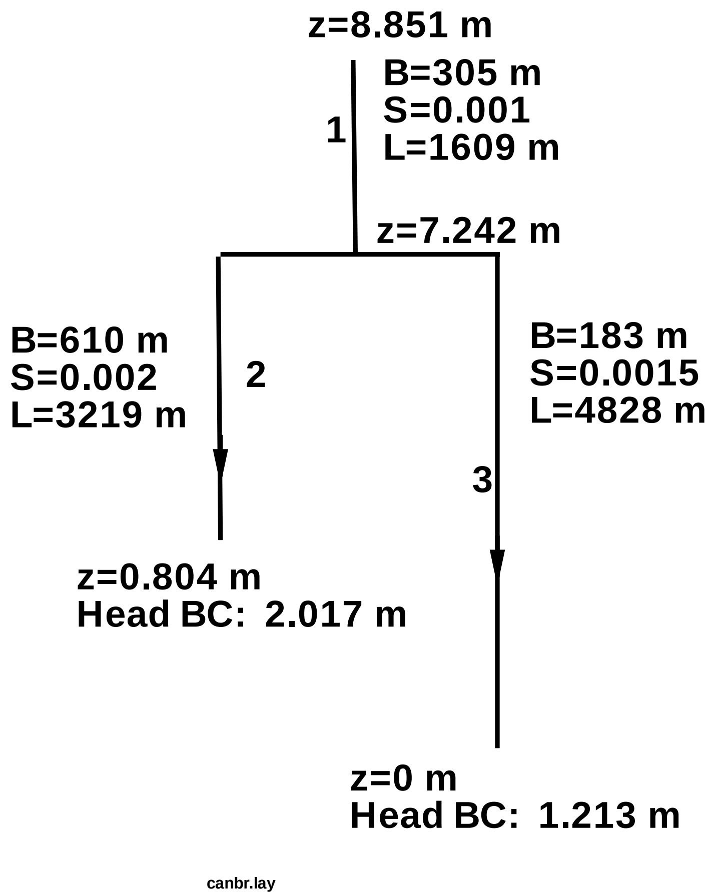

Fig 6.3: Canal configuration used for test 2.

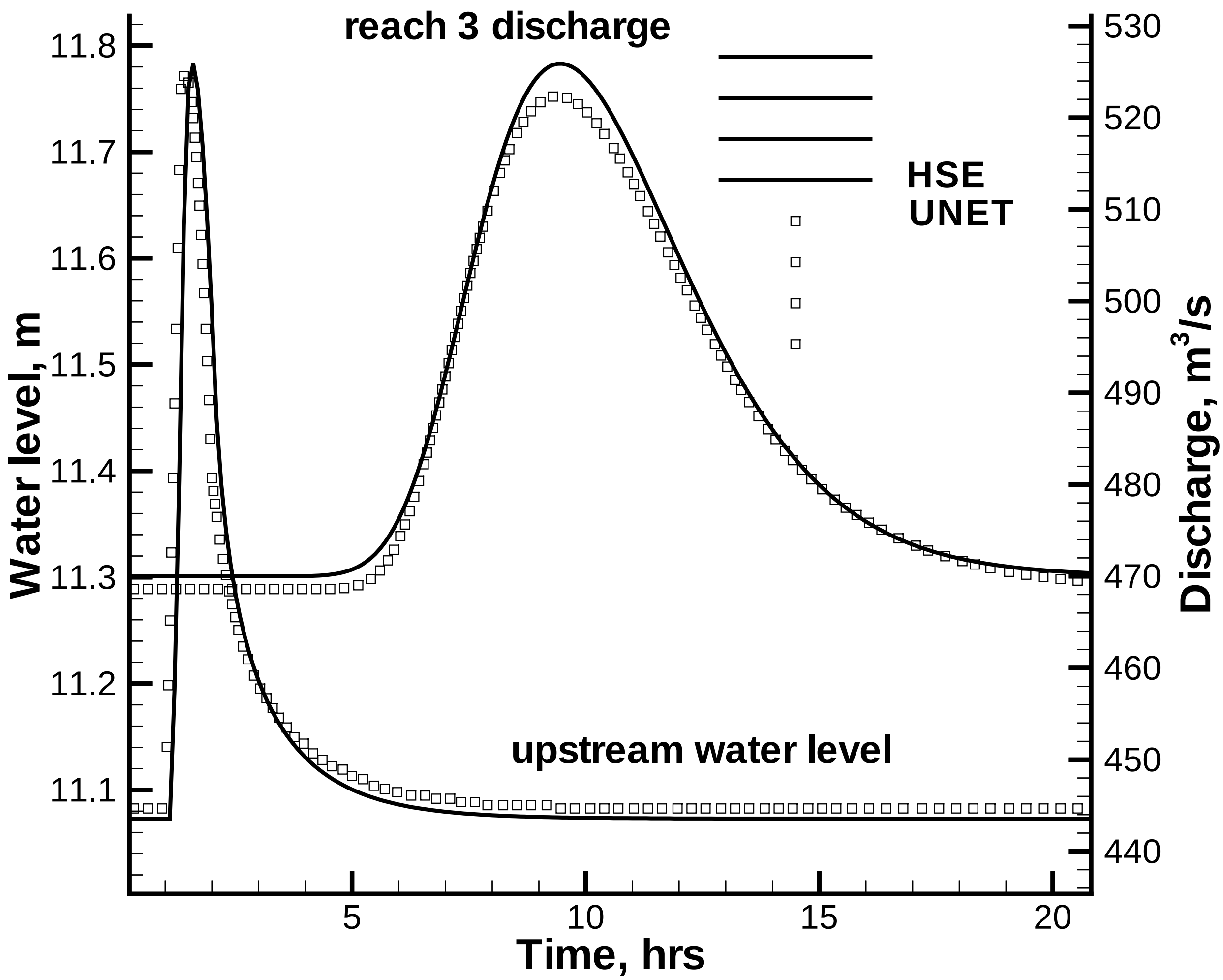

Fig 6.4: Comparison of HSE and UNET models.

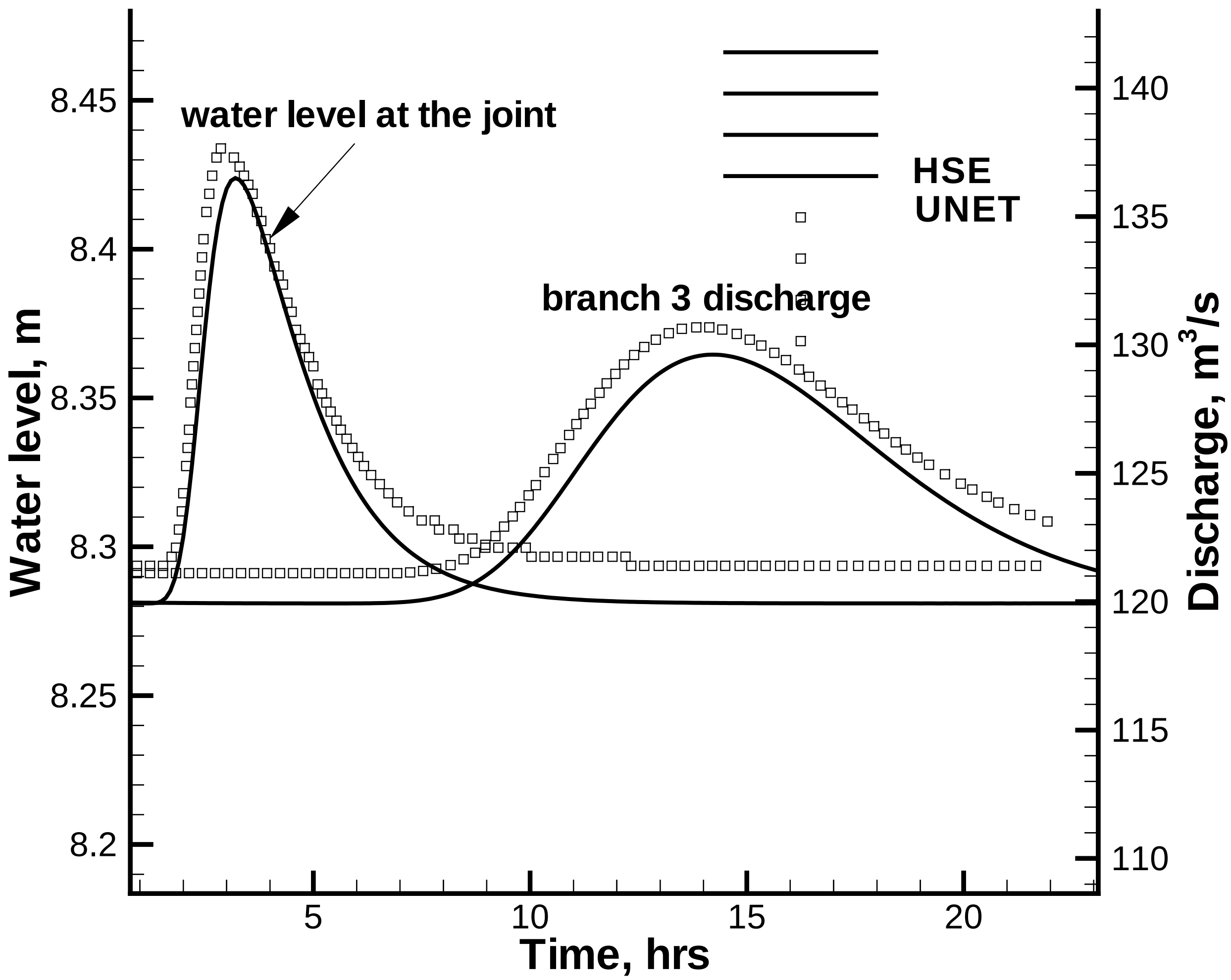

Fig 6.5: Comparison of HSE and UNET models.

Chapter 7

Linear equation solvers

When using implicit methods, linear equation solvers are required to solve for the final water level. Solution of general linear equation solvers is time consuming. However, if the physical domain is elongated for example, it is possible to obtain faster solutions using banded matrix solvers if the cells are numbered carefully to minimize the bandwidth. In the current model, cells interact only with a limited number of other cells in the neighborhood, and therefore the system of equations is generally sparse. Research work on sparse solvers has recently become active because of their use in many numerical and network models. A number of solvers are available from major research labs. The Argonne National lab, the Lawrence Livermore lab, NASA, IBM, Silicon Graphics and many other large scale software developers support sparse equation solvers.

Both sparse and dense systems of linear equations can be solved using direct and indirect methods. Direct method generally include elimination methods which do not need iterations. A commonly used iterative method uses optimization, as in the case of the conjugate gradient method. When using conjugate gradient methods, the system of linear equation is expressed as $\mathbf{A}.\mathrm{x} = \mathbf{b}$, and an approximate solution of x is obtained by minimizing $||\mathbf{r}||_2$ in which $\mathbf{r} = \mathbf{b} - \mathbf{A}.\mathrm{x} =$ residual vector. If x^* is the proper solution such that $\mathbf{A}.\mathrm{x}^* = \mathbf{b}$, the function to be minimized can be expressed as

$$f(\mathrm{x}) = ||\mathbf{A}.\mathrm{x} - \mathbf{b}|| = (\mathrm{x} - \mathrm{x}^*)^T \mathbf{C}(\mathrm{x} - \mathrm{x}^*) \tag{7.1}$$

in which, $\mathbf{C} = \mathbf{A}^{\mathbf{T}}.\mathbf{A}$. When using a gradient method such as the Gauss Newton method, the gradient is computed as

$$\nabla f(\mathbf{x}) = 2\mathbf{C}.(\mathbf{x} - \mathbf{x}^*) = -2\mathbf{r} \tag{7.2}$$

In the gradient method, iterations are carried out using

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t\nabla f(\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} + 2t\mathbf{r}^{(k)} \tag{7.3}$$

in which k is the iteration number. Value of the scalar t required to minimize f can be substituted to the above equation to obtain the following expression for $\mathbf{x}^{(k+1)}$.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\mathbf{r}^{(k)T}.\mathbf{r}^{(k)}}{\mathbf{r}^{(k)T}.\mathbf{C}.\mathbf{r}^{(k)}}\mathbf{r}^{(k)} \tag{7.4}$$

The rate of convergence of $\mathbf{r}$ is related to eigenvalues of $\mathbf{C}$ using the following equation.

$$||\mathbf{r}^{(k+1)}||_2^2 \leq \left(\frac{M-m}{M+m}\right)^2 ||\mathbf{r}^{(k)}||_2^2 \tag{7.5}$$

in which M and m are the largest and smallest eigenvalues of the positive definite matrix $\mathbf{C}$. The rate of convergence may be improved when the cell sizes and cell conductivity properties are nearly uniform throughout the physical domain. Reduction of the time step may be a last resort to achieve the same goal.

The PetSc solver developed by the Argonne National lab and the SLAP solver developed by the Lawrence Livermore Lab take advantage of the iterative method described earlier. Higher level routines in the package require the storage of only the nonzero elements of $\mathbf{A}$ and their positions. Even this can be avoided if the user writes his own subroutine for multiplying the matrix times a vector and calls the lower-level iterative routines in the package.

7.0.1 SLAP 2.0 Sparse Linear Algebra Package

This package developed by the Lawrence Livermore Lab contains routines for the iterative solution of symmetric and non-symmetric positive definite and positive semi-definite

Table 6.1: Model run times with the SLAP 2.0 sparse solver.

Method	Pre Proc. (s)	Each step (s)	NITER	DT (days)	H (m)	Comm.
Explicit	0.8	3.8		0.5	0.441	Max. DT
(1) Jacobi	100.2	2.3	30	0.5		Diverged
(2) Gauss-Seidel	99.2	2.1-2.2	7-17	1.0	0.446	Conver.
(2) Gauss-Seidel	98.8	2.3	21	6.0	0.465	"
(3) Incomplete LU Iter. Refinement	99.7	2.5	13-9	6.0	0.465	"
(4) Diagonally scaled Conj. Grad.	98.8	5.9	97	6.0	0.465	"
(5) Incomplete LU Conj. Grad.	99.0	2.6	15	6.0	0.465	"
(6) Diagonally scaled Biconj. Grad.	99.8	2.6	20,17	6.0	0.465	"
(7) Incomplete LU Biconj. Grad.	100.3	2.4	7	6.0	0.465	"
(8) Diagonally scaled preconditioned Bi-Conj. grad.	100.7	3.1	18,10	6.0	0.465	"
(9) Incomplete LU Biconj. Grad.	100.4	2.34	4	6.0	0.465	"
(10) Diagonally scaled Orthomin	100.4	2.34	27	6.0	0.465	"
(11) Incomplete LU Orthomin	100.1	2.22	10	6.0	0.465	"
(11) Diagonally scaled generalized min. residual						Error
(12) Incomplete LU generalized min. res.						Error

linear systems. Included in this package are core routines to do iterative refinement iteration, preconditioned Conjugate Gradient iteration, preconditioned biConjugate gradient iteration, Preconditioned biConjugate gradient squared iteration, orthomin iteration and generalized Minimum Residual iteration. The authors are Dr. Mark K. Seager, Lawrence Livermore National Lab., and Dr. Anne Greenbaum of the Courant Institute of Mathematical Sciences. Package was last updated in 1989, and is available from `netlib` of the Internet.

7.0.2 Test runs with the SLAP2.0 solver

A version of the model with 1250 triangular grids was tested to compare the run times of various iterative solvers. A 12 day simulation with 6 day time steps was used in the fully implicit mode. An explicit run could not be made with more than 0.5 day time steps. Table 6.1 shows the run times of the model obtained using a Sparc 20 machine.

7.0.3 Which Method To Use

In solving a large sparse linear system $Ax = b$ using an iterative method, it is not necessary to actually store the matrix A. Rather, what is needed is a procedure for multiplying the matrix A times a given vector y to obtain the matrix-vector product, Ay. SLAP has been written to take advantage of this fact. The higher level routines in the package require storage only of the nonzero elements of A (and their positions), and even this can be avoided, if the user writes his own subroutine for multiplying the matrix times a vector and calls the lower-level iterative routines in the package.

If the matrix A is ill-conditioned, then most iterative methods will be slow to converge (if they converge at all!). To improve the convergence rate, one may use a "matrix splitting," or, "preconditioning matrix," say, M. It is then necessary to solve, at each iteration, a linear system with coefficient matrix M. A good preconditioner M should have two properties: (1) M should "approximate" A, in the sense that the matrix inv(M)*A (or some variant thereof) is better conditioned than the original matrix A; and (2) linear systems with coefficient matrix M should be much easier to solve than the original system with coefficient matrix A. Preconditioning routines in the SLAP package are separate from the iterative routines, so that any of the preconditioners provided in the package, or one that the user codes himself, can be used with any of the iterative routines.

7.0.4 Choice of the preconditioner

If you willing to live with either the SLAP Triad or Column matrix data structure of SLAP, you can then choose one of two types of preconditioners to use: diagonal scaling or incomplete factorization. To choose between these two methods requires knowing something about the computer you're going to run these codes on and how well incomplete factorization approximates the inverse of your matrix.

Let's suppose you have a scalar machine. Then, unless the incomplete factorization is very poor, this is *GENERALLY* the method to choose. It will reduce the number of iterations significantly and is not all that expensive to compute. So if you have just one linear system to solve and "just want to get the job done" then try incomplete factorization first. If you are thinking of integrating some SLAP iterative method into your favorite "production code" then try incomplete factorization first, but also check to see that diagonal scaling is indeed slower for a large sample of test problems.

If your matrix is symmetric then you would want to use one of the symmetric system solvers. If your system is also positive definite, (Ax,x) (Ax dot product with x) is positive for all non-zero vectors x, then use Conjugate Gradient (SCG, SSDCG, SSICSG) methods. If you're not sure it's SPD (symmetric and Positive Definite) then try SCG anyway and if it works, fine. If you're sure your matrix is not positive definite then you may want to try the iterative refinement methods (SIR) or the GMRES code (SGMRES) if SIR converges too slowly.

If the matrix is symmetric, you are working in an area of active research in numerical analysis and there are new strategies being developed. Consequently take the following advice with a grain of salt. If you matrix is positive definite, (Ax,x) (Ax dot product with x is positive for all non-zero vectors x), then you can use any of the methods for nonsymmetric systems (Orthomin, GMRES, BiConjugate Gradient, BiConjugate Gradient Squared and Conjugate Gradient applied to the normal equations). If your system is not too ill conditioned then try BiConjugate Gradient Squared (BCGS) or GMRES (SGMRES). Both of these methods converge very quickly and do not require A' or M' (' denotes transpose). SGMRES does require some additional storage. If the system is very ill conditioned or nearly positive indefinite ((Ax,x) is positive, but may be very small), then GMRES should be the first choice, but try the other methods if you have to fine tune the solution process for a "production code". If you have a great preconditioner for the normal equations (i.e., M is an approximation to the inverse of AA' rather than just A) then this is not a bad route to travel. Old wisdom would say that the normal equations are a disaster (since it squares the condition number of the system and SCG convergence is linked to this number of infamy), but some preconditioners (like incomplete factorization) can reduce the condition number back below that of the original system.

REFERENCES

Strang, G. (1968). "On the construction and comparison of difference schemes", *SIAM Journal of Numerical Analysis*, 5(3), 506-517.

Szidarovszky, F., and Yakowitz, S., (1978). *Principles and procedures of numerical analysis*, Plenum Press, New York.

Wang, Herbert. (1982). *Introduction to groundwater modeling*, W. H. Freeman Co.

Lou, J. Z. (1997). "Parallel multigrid algorithm and code for computations of incompressible flow", *NASA Tech Briefs*, April 1997, p-73.

Viessman, W., Knapp, J. W., Lewis, G. L., and Harbaugh, T. E. (1977). *Introduction to Hydrology*, Harper & Row Publishers, New York.

Appendix A

Benchmark test 1 for overland flow

Benchmark test are used during the development of the model to monitor the model results during every stage of development. For purposes of monitoring, the following intermediate and final results are recorded for a number of test cases so that future runs can be checked against them. In the first benchmark test, an axisymmetric test problem is used. Future benchmark tests will be designed to monitor the execution times of model runs. For the first benchmark test problem, all units are considered to be in SI. Other parameters assumed are; the Manning's coefficient = 1.0, STOL = 1.0E-9, VLARG = 1.0E25.

Benchmark test 1:

The first benchmark test is aimed at checking the wall numbers, matrices, and solutions of the linear equations. The input data set for the problem is shown below. Figure 2.1 shows the grid for this data set.

```
TT          NT   ALP METH IOPG
10000      10    0.5  5    1
NE (elements), ND (nodes) NITER
18   16   1
NODE(I,K), K=1,4  (Nodal connectivity)
    1    2    6    6
    2    3    7    7
    3    4    8    8
    5    6   10   10
    6    7   11   11
    7    8   12   12
    9   10   14   14
   10   11   15   15
   11   12   16   16
    1    6    5    5
    2    7    6    6
    3    8    7    7
    5   10    9    9
    6   11   10   10
    7   12   11   11
```

```
    9   14   13   13
   10   15   14   14
   11   16   15   15
NN X Y;  ND values of nodal coordinates
    1          0.000      15000.000
    2       5000.000      15000.000
    3      10000.000      15000.000
    4      15000.000      15000.000
    5          0.000      10000.000
    6       5000.000      10000.000
    7      10000.000      10000.000
    8      15000.000      10000.000
    9          0.000       5000.000
   10       5000.000       5000.000
   11      10000.000       5000.000
   12      15000.000       5000.000
   13          0.000          0.000
   14       5000.000          0.000
   15      10000.000          0.000
   16      15000.000          0.000
NN Z BMAN;  NE values of topos and initial water levels
   1    500.000    1.00000
   2    500.000    1.00000
   3    500.000    1.00000
   4    500.000    1.00000
   5    500.000    1.00000
   6    500.000    1.00000
   7    500.000    1.00000
   8    500.000    1.00000
   9    500.000    1.00000
   10    500.000    1.00000
   11    500.000    1.00000
   12    500.000    1.00000
   13    500.000    1.00000
   14    500.000    1.00000
   15    500.000    1.00000
   16    500.000    1.00000
   17    500.000    1.00000
   18    500.000    1.00000
NB No of externally imposed wall bnds
0
NN N3 N4 N5 N6
NG No of externally imposed GW wall bnds
0
NN N3 N4 N5 N6
NH No. of externally imposed head bnds
0
NN N1 ICUR
NQ No of source/sink
0
NN N1 ICUR
```

```
NF Number of flow lines to be defined
0
NN NO - Flow line no., number of pairs; followed by pairs themselves
```

The following intermediate results include the wall number assignments.

NN	N1	N2	N3	N4	N5	N6	N7	N8
1	1	10	1	6	2	0	2	0
2	1	11	2	6	2	0	2	0
3	2	11	2	7	2	0	2	0
4	2	12	3	7	2	0	2	0
5	3	12	3	8	2	0	2	0
6	4	10	5	6	2	0	2	0
7	4	13	5	10	2	0	2	0
8	4	14	6	10	1	0	1	0
9	5	11	6	7	1	0	1	0
10	5	14	6	11	1	0	1	0
11	5	15	7	11	1	0	1	0
12	6	12	7	8	2	0	2	0
13	6	15	7	12	2	0	2	0
14	7	13	9	10	2	0	2	0
15	7	16	9	14	2	0	2	0
16	7	17	10	14	2	0	2	0
17	8	14	10	11	1	0	1	0
18	8	17	10	15	2	0	2	0
19	8	18	11	15	2	0	2	0
20	9	15	11	12	2	0	2	0
21	9	18	11	16	2	0	2	0
22	1	0	1	2	0	0	0	0
23	2	0	2	3	0	0	0	0
24	3	0	3	4	0	0	0	0
25	3	0	4	8	0	0	0	0
26	6	0	8	12	0	0	0	0
27	9	0	12	16	0	0	0	0
28	10	0	1	5	0	0	0	0
29	13	0	5	9	0	0	0	0
30	16	0	13	14	0	0	0	0
31	16	0	9	13	0	0	0	0
32	17	0	14	15	0	0	0	0
33	18	0	15	16	0	0	0	0

```
NN     = wall number
N1, N2 = cells opposite the wall
N3, N4 = nodes defining the cell
N5     = overland flow wall type
N6     = overland flow wall sequence number
N7     = groundwater wall type
N8     = groundwater wall sequence number
```

The output from within the time step loop consist of the K matrix. The following table shows the non-zero values of the K matrix during the first three time steps.

Row	Col	K (setp1)	K (step 2)	K (step 3)
1	1	0.00000E+00	-0.79067E+03	-0.61761E+03
1	10	0.00000E+00	0.00000E+00	0.00000E+00
1	11	0.00000E+00	0.79067E+03	0.61761E+03
2	2	0.00000E+00	-0.14060E+04	-0.31711E+04
2	11	0.00000E+00	0.14060E+04	0.11200E+04
2	12	0.00000E+00	0.00000E+00	0.20511E+04
3	3	0.00000E+00	0.00000E+00	0.00000E+00
3	12	0.00000E+00	0.00000E+00	0.00000E+00
4	4	-0.13887E+03	-0.23370E+04	-0.18791E+04
4	10	0.00000E+00	0.79066E+03	0.61761E+03
4	13	0.00000E+00	0.14060E+04	0.11200E+04
4	14	0.13887E+03	0.14033E+03	0.14147E+03
5	5	-0.27773E+03	-0.28067E+03	-0.28294E+03
5	11	0.13887E+03	0.14033E+03	0.14147E+03
5	14	0.00000E+00	0.00000E+00	0.00000E+00
5	15	0.13887E+03	0.14033E+03	0.14147E+03
6	6	0.00000E+00	-0.14060E+04	-0.31711E+04
6	12	0.00000E+00	0.00000E+00	0.20511E+04
6	15	0.00000E+00	0.14060E+04	0.11200E+04
7	7	0.00000E+00	0.00000E+00	-0.41021E+04
7	13	0.00000E+00	0.00000E+00	0.20511E+04
7	16	0.00000E+00	0.00000E+00	0.00000E+00
7	17	0.00000E+00	0.00000E+00	0.20511E+04
8	8	-0.13887E+03	-0.23370E+04	-0.18791E+04
8	14	0.13887E+03	0.14033E+03	0.14147E+03
8	17	0.00000E+00	0.14060E+04	0.11200E+04
8	18	0.00000E+00	0.79066E+03	0.61761E+03
9	9	0.00000E+00	-0.79067E+03	-0.61761E+03
9	15	0.00000E+00	0.79067E+03	0.61761E+03
9	18	0.00000E+00	0.00000E+00	0.00000E+00

```
10   1  0.00000E+00  0.00000E+00  0.00000E+00
10   4  0.00000E+00  0.79066E+03  0.61761E+03
10  10  0.00000E+00 -0.79066E+03 -0.61761E+03
11   1  0.00000E+00  0.79067E+03  0.61761E+03
11   2  0.00000E+00  0.14060E+04  0.11200E+04
11   5  0.13887E+03  0.14033E+03  0.14147E+03
11  11 -0.13887E+03 -0.23370E+04 -0.18791E+04
12   2  0.00000E+00  0.00000E+00  0.20511E+04
12   3  0.00000E+00  0.00000E+00  0.00000E+00
12   6  0.00000E+00  0.00000E+00  0.20511E+04
12  12  0.00000E+00  0.00000E+00 -0.41021E+04
13   4  0.00000E+00  0.14060E+04  0.11200E+04
13   7  0.00000E+00  0.00000E+00  0.20511E+04
13  13  0.00000E+00 -0.14060E+04 -0.31711E+04
14   4  0.13887E+03  0.14033E+03  0.14147E+03
14   5  0.00000E+00  0.00000E+00  0.00000E+00
14   8  0.13887E+03  0.14033E+03  0.14147E+03
14  14 -0.27773E+03 -0.28067E+03 -0.28294E+03
15   5  0.13887E+03  0.14033E+03  0.14147E+03
15   6  0.00000E+00  0.14060E+04  0.11200E+04
15   9  0.00000E+00  0.79067E+03  0.61761E+03
15  15 -0.13887E+03 -0.23370E+04 -0.18791E+04
16   7  0.00000E+00  0.00000E+00  0.00000E+00
16  16  0.00000E+00  0.00000E+00  0.00000E+00
17   7  0.00000E+00  0.00000E+00  0.20511E+04
17   8  0.00000E+00  0.14060E+04  0.11200E+04
17  17  0.00000E+00 -0.14060E+04 -0.31711E+04
18   8  0.00000E+00  0.79066E+03  0.61761E+03
18   9  0.00000E+00  0.00000E+00  0.00000E+00
18  18  0.00000E+00 -0.79066E+03 -0.61761E+03
```

Intermediate results also contain the right hand sides and the solution vector. If the time step is small enough that even the explicit option is stable, solution can be compared with the RHS/cell area. For cell 5 for example, solution at step 1 is 0.02185 m, and the RHS/Area is 0.0222 m, which are very close. This feature can be used to debug the model.

Cell	RHS (step1)	RHS (step2)	RHS (step3)	Sol (step1)	Sol (step2)	sol (step3)
1	0.00000E+00	0.86398E+04	0.11228E+05	0.00000E+00	0.92056E-03	0.10587E-02
2	0.00000E+00	0.15364E+05	0.16323E+05	0.00000E+00	0.15989E-02	0.15074E-02
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.13887E+06	0.11173E+06	0.10182E+06	0.10927E-01	0.81733E-02	0.75535E-02
5	-0.27773E+06	-0.27147E+06	-0.26531E+06	-0.21855E-01	-0.21386E-01	-0.20902E-01
6	0.00000E+00	0.15364E+05	0.16323E+05	0.00000E+00	0.15989E-02	0.15074E-02
7	0.00000E+00	0.00000E+00	0.65586E+04	0.00000E+00	0.00000E+00	0.66321E-03
8	0.13887E+06	0.11173E+06	0.10182E+06	0.10927E-01	0.81733E-02	0.75535E-02
9	0.00000E+00	0.86398E+04	0.11228E+05	0.00000E+00	0.92056E-03	0.10587E-02
10	0.00000E+00	0.86398E+04	0.11228E+05	0.00000E+00	0.92056E-03	0.10587E-02
11	0.13887E+06	0.11173E+06	0.10182E+06	0.10927E-01	0.81733E-02	0.75535E-02
12	0.00000E+00	0.00000E+00	0.65586E+04	0.00000E+00	0.00000E+00	0.66321E-03
13	0.00000E+00	0.15364E+05	0.16322E+05	0.00000E+00	0.15989E-02	0.15074E-02
14	-0.27773E+06	-0.27147E+06	-0.26531E+06	-0.21855E-01	-0.21386E-01	-0.20902E-01
15	0.13887E+06	0.11173E+06	0.10182E+06	0.10927E-01	0.81733E-02	0.75535E-02
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.15364E+05	0.16322E+05	0.00000E+00	0.15989E-02	0.15074E-02
18	0.00000E+00	0.86398E+04	0.11228E+05	0.00000E+00	0.92056E-03	0.10587E-02

The following table shows the water levels at 1000 s time intervals for a number of cells.

Time (s)	Cell 5	Cell 11	Cell 2	Cell 12	Cell 3	Cell 1
0.00000	2.00000	1.00000	1.00000	1.00000	1.00000	1.00000
1000.00000	1.97815	1.01093	1.00000	1.00000	1.00000	1.00000
2000.00000	1.95676	1.01910	1.00160	1.00000	1.00000	1.00092
3000.00000	1.93586	1.02665	1.00311	1.00066	1.00000	1.00198
4000.00000	1.91543	1.03365	1.00474	1.00114	1.00033	1.00316

5000.00000	1.89546	1.04014	1.00650	1.00169	1.00070	1.00444
6000.00000	1.87594	1.04617	1.00834	1.00231	1.00111	1.00581
7000.00000	1.85687	1.05178	1.01026	1.00300	1.00155	1.00725
8000.00000	1.83822	1.05700	1.01224	1.00374	1.00204	1.00876
9000.00000	1.81999	1.06185	1.01428	1.00453	1.00256	1.01033
10000.00000	1.80217	1.06636	1.01635	1.00537	1.00312	1.01195

Appendix B

Benchmark test 1 for canal network flow

The following is a sample input data set for the 1-D network problem shown in Figure 5.1. The intermediate results for the test will be included later once the debugging is complete. A description of the data set is given in Chapter 5.

```
NL NJ TT      NT  ALP  METH NITER
9  8  3600. 36  0.6   9     1
NN  Segs. LINK(I,K),K=1,LINK(I,1)
 1   1  1
 2   2  3  4
 3   3  1  2  3
 4   3  2  5  9
 5   3  4  9  7
 6   2  7  8
 7   2  5  6
 8   2  6  8
NN DATUM  WID   SLOP AREA     LENGTH   MAN
 1  6.6548 6.096 0.0  11.1484  804.6   0.02
 2  5.4483 6.096 0.0  11.1484  804.6   0.02
 3  4.2418 6.096 0.0  11.1484  804.6   0.02
 4  3.0353 6.096 0.0  11.1484  804.6   0.02
 5  3.2032 6.096 0.0  11.1484  804.6   0.02
 6  3.4374 6.096 0.0  11.1484  804.6   0.02
 7  2.9374 6.096 0.0  11.1484  804.6   0.02
 8  5.8256 6.096 0.0  11.1484  804.6   0.02
NCB No. of flowpair boundaries
```

```
 1
NN N1 N2 TYP ICUR
 1  4  3  3   1
  0.0015
NCH Head boundaries
 0
NN N1 ICUR
NCQ Flow boundaries
 1
NN N1 ICUR
 1  1  1
```

Benchmark test 2 for canal network flow

The following benchmark test was used to check intermediate results of the test in Viesmann's text. The channel is divided only into 5 sections so that the output is not too huge for the debugger. The input data file is as shown below.

```
NL NJ TT     NT  ALP  METH NITER
5  6  3600. 12   0.5   9     1
NN  Segs  LINK(I,K),K=1,LINK(I,1)
 1   1  1
 2   2  1  2
 3   2  2  3
 4   2  3  4
 5   2  4  5
 6   1  5
NN DATUM   WID   SLOP AREA     LENGTH   MAN
```

```
 1  6.6568 6.096 0.0  11.1484  804.6   0.02
 2  5.4498 6.096 0.0  11.1484  804.6   0.02
 3  4.2428 6.096 0.0  11.1484  804.6   0.02
 4  3.0358 6.096 0.0  11.1484  804.6   0.02
 5  1.8288 6.096 0.0  11.1484  804.6   0.02
NCB No. of flowpair boundaries
 1
NN N1 N2 TYP ICUR
 1  5  4  3   1
  0.0015
NCH Head boundaries
 0
NN N1 ICUR
NCQ Flow boundaries
 1
NN N1 ICUR
 1  1  1
```

As shown in the data file, unform flow boundary is of wall boundary type 3, across segment 5 and 4. There is one parameter involved with the boundary. This parameter is the slope of the the uniform flow, which is 0.0015 in this case. The time step is 300 s; ALP=0.5. The initial conditions are given with the following data file.

```
   6.6568
   5.4498
   4.2428
   3.0358
   1.8288
```

and the output is given as

Time (s)	H1	H3	H5	Q12	Q23	Q45
300.000	6.65660	5.44974	4.24279	23.5989	23.6008	23.6010
600.000	7.01213	5.55518	4.27473	28.4883	25.0380	23.9929
900.000	7.49610	5.85984	4.41536	34.0691	28.8599	25.6445
1200.00	7.96649	6.31984	4.72373	40.8121	34.8165	29.3736
1500.00	8.36481	6.82307	5.18673	48.4715	42.4144	35.6180
1800.00	8.26191	7.07059	5.62911	47.2384	46.9648	42.6789
2100.00	8.03579	6.99561	5.80756	48.5442	48.1602	47.4009
2400.00	7.79554	6.78641	5.71690	47.4425	48.8837	49.7458
2700.00	7.56304	6.54896	5.50939	44.0237	47.1199	49.5885
3000.00	7.34041	6.31762	5.27944	39.7949	43.3786	46.6690
3300.00	7.12210	6.09672	5.05606	35.4700	39.0834	42.5543
3600.00	6.90236	5.88151	4.84201	31.2498	34.8184	38.2685

Appendix C

Selected subroutines used in the canal network module

CLINK

This routine is used to read all the static data for the model except for the boundary condition data.

CANWK

This routine is used to create flow pairs or flow walls between canal sections, and number them 1, 2, etc., NPAIR. If there are only two segments, there is only one pair. For a joint with three segments, there are 3 pairs. In general, there are ${}^{n}_{2}C$ pairs, in which i is the number of branches. Each pair I has segments IPAIR(I,1), and IPAIR(I,2), and is defined by type IPAIR(I,3). All canal types are also stored in IXL(I) to identify them as end segments or middle segments.

CSTBN

This will change the flow pair type to new types such as structure types when structure type internal boundary conditions are specified.

UNIFC

This is where uniform flow boundary condition is specified. Uniform flow is achieved by assigning the last two river segments to reflect the slope of the uniform flow. In the code, the end river segment N1 is positioned with respect to the previous segment, and N th row of the matrix is modified.

CFORM

In CFORM, wall resistances of flow pairs are computed. The method used for joints with 2 segments is not the same as the method used for joints with more than 2 segments. Matrix values are computed to get ready for the implicit solution.

BOUNC

Wall resistance across structures is computed, and added to the matrix.

COURC

Source term information is computed here. The boundary flows as well as branch inflows are considered as part of the flow.

Appendix D

Definition of variables in the overland flow module

Table C.1: Definition of input variables in the overland flow module.

`ALP`	is an input parameter. A value of 0 selects the explicit solution, 1 selects the implicit solution, and 0.5 or higher selects fairly stable solutions suitable for most simulations.
`BMAN(i)`	Manning's roughness coefficient of cell i.
`COND(i)`	average conductivity of the soil in cell i.
`CONS(i,j)`	constants defining structure boundary conditions. $i =$ the sequence number of the structure provided as input by the used counting 1, 2, 3, n, n+1; $j =$ counter for the constants, ie, j=0 refers to C_0 etc.
`CORN(i)`	coordinates of node i in complex form.
`DET(i)`	Detention depth of cell i.
`DT`	in DTRAD gives the time interval at which boundary condition time series data is provided in data file `qhbnd.dat`.
`DTO`	in subroutine MAIN defines the time interval at which the output is printed.
`HO(i)`	Water level in cell i at previous time step. Initial conditions are provided as input data.
`HBND(i)`	water level boundary conditions, stored in an array. i are the sequence numbers defined as ICUR.
`IHBND(i,j)`	definition of head boundary condition type; $i = 1, 2, \ldots NH$ gives the boundary condition sequence number; $j = 1$ gives the cell number on which it is imposed; $j = 2$ gives the curve number or sequence number of the time series head boundary condition HBND.

Table C.2: Definition of variables in the overland flow module.

IQBND(i,j)	definition of source/sink (pumping, inflow etc.) boundary condition type; $i = 1, 2, \ldots NQ$ gives the boundary conditions sequence number; $j = 1$ gives the cell number on which it is imposed; $j = 2$ gives the curve number or sequence number of the time series head boundary condition HBND.
IOPG	Variable used to select the type of run; =1 if it is an overland flow only case; =2 if it is an ground water only case; =3 if both overland and ground water flow are considered.
METH	is an input parameter used to select the sparse linear solution method. Use a value of 1-13 in the case of SLAP. Values of 6 and 7 are found to be suitable for most cases for SLAP.
NB	number of overland flow boundary conditions attached to walls.
ND	Number of nodes
NE	Number of cells
NH	number of head boundary conditions.
NG	number of ground water flow boundary conditions attached to walls.
NITER	Number of iterations carried out within a time step until the matrix K used is accurate for the time step.
NODE(i,j)	Node numbers $j = 1, 2, ..6$ defining element numbers around node i. $j = 7$ is reserved to store the number of nodes. Third and fourth node numbers are repeated in triangles. Polygons up to hexagons can be stored without re-dimensioning.
NQ	number of source/sink boundary conditions attached to cells.
NT	is an input parameter asking for the number of time steps
NWAL(i,5)	overland flow boundary condition types attached to walls; $i =$ wall number; j= 0 is a no flow; $j = 1$ is 2-D flow using line integrated flow vector, $j = 2$ is direct flow equation, $j = 4$ is weir boundary, $>$ 4 yet undefined structure boundaries.

Table C.3: Definition of variables in the overland flow module.

`NWAL(i,7)`	ground water flow boundary condition types attached to walls; i = wall number; j= 0 is a no flow; $j = 1$ is 2-D flow using line integrated flow vector, $j = 2$ is direct flow equation, $j >$ 4 yet undefined structure boundaries.
`QBND(i)`	source sink type boundary condition time series i. i is the curve number that corresponds to the sequence number defined in ICURV at the input.
`SG(I)`	specific yield of soil in cell i.
`THET`	Fraction of the total area which is wet.
`TT`	is an input parameter, asking for the total simulation time in sec.
`TOLH`	in subroutine MAIN, TOLH gives a tolerance limit for the iteration loop for refining K to exit.
`TOL`	in KFORM is the cutoff ΔH below which, K computed using the Manning's equation is considered too large.
`Z(i)`	average elevation in cell i; If the elevation is affecting the wet area, Z(i) is the elevation at which area = 0.
`ZB(i)`	elevation of the bottom of the aquifer.
`ZU(i)`	elevation of the cell when the entire cell jest gets wet.

Table C.4: Table showing the definition of some internal variables.

Name	Definition
A(i)	The matrix solved by the implicit method (self)
ALP	Time weighing factor; = 0 means explicit and =1 means implicit (input).
AREA(i)	Area of cell i
B(i)	Right hand side of the matrix solution scheme.
CENT(i)	Coordinates of the cell centroid of cell i in complex form (self)
DT	Time step (input)
DTO	Time interval at which the output is saved (input)
H1(i)	Water level in cell i at current time step (active state variable)
HBND(i)	Current head value from time series i (self)
HCL(i)	Water stage at node i if needed (self)
IBC(i,1)	1 makes cell i a boundary cell for overland flow; 0 otherwise.
IBC(i,2)	1 makes cell i a boundary cell for groundwater flow; 0 otherwise.
ID(i,j)	Storage location in the 1-D matrix of a non-zero value that belong in row i, column j.
METH	The linear solution method used, (1-13 for slap) (input)
NCL(i,j)	Cell numbers clustering around node i, in anti-clockwise direction (self generated). $j = 1$ is used to store the number of cells around the nodes. NCL(i,j),j=2,.. are used to store the cell numbers.
NELT	Number of non-zero elements.
NT	Number of time steps (Input)
NW	Number of internal cell walls (self)
NWAL(i,j)	Cell wall information for walls $i = 1, NW$. $J = 1$ and $J = 2$ give the cell numbers on both sides of the wall. $J = 3$ and $J = 4$ give nodes defining the wall i in ascending order. $j = 5$ is reserved to give the type of the cell boundary. $j = 6$ gives the specific boundary condition curve number.

Table C.5: Table showing the definition of some internal variables.

Name	Definition
QBND(i)	Current discharge rate at time series i.
TIME	Time elapsed in the simulation (self)
TT	Total run time
UX(i), UY(i)	overland flow velocities in X and Y directions (self)
VX(i), VY(i)	groundwater flow velocities in X and Y directions (self)
XKG(i)	groundwater conductivity of wall i.
XKO(i)	overland conductivity of wall i.
XLL(i,1)	Length of cell wall i (self generated).
XLL(i,2)	Distance from cell wall to circumcenter of cell N1 (self).
XLL(i,3)	Distance from cell wall to circumcenter of cell N2 (self).
XLC(i),XLS(i)	Direction cosines of the slopes of the outward normal to cell wall i (self)
ZBCL(i)	Average elevation of the bottom of aquifer at a node (self).
ZCL(i)	Average ground elevation at a node (self)

Appendix D

Definition of variables in the canal flow module

Table D.1: Definition of input variables in the canal flow module.

`A0(i)`	The K matrix in 1-D storage mode, for the previous time step. This information is to be used in the iterative mode.
`A1(i)`	The K matrix in 1-D storage mode, for the current time step.
`AA(i)`	A(i) matrix saved to be used in case of instability.
`B(i)`	The right hand side vector.
`CAREA(i)`	Area of canal section i.
`CBAR(i)`	Canal area below the datum.
`CBWI(i)`	Canal width at datum.
`CBOT(i)`	Elevation of the canal bottom below which there is no flow.
`CKNOD(i)`	K value of the node (i).
`CMAN(i)`	Manning's roughness of canal segment.
`CON(i,j)`	Constants used to define structures. i gives the sequence numbers attached to the structures, j gives the constants. In the case of uniform flow, CON(i,1) gives the uniform flow slope.
`CSLOP(i)`	Bank slope of the canal.
`CWIDT(i)`	Width of canal i.
`DHC(i)`	ΔH
`HC0(i)`	Head of canal section i at timestep n.
`HC1(i)`	Head of canal section i at timestep $n+1$.
`HCBAS(i)`	Datum of canal at which area is provided as CBAR(i).
`HCBND(i)`	Head boundary condition given as time series i.

`IHBNC(i,j)`	Array providing head boundary conditions. i gives the head boundary condition sequence number, marked 1,2,3,etc.. $j = 1$ gives the canal segment number; $j = 2$ gives the sequence number of the input time series data file related to the canal segment.
`IQBNC(i,j)`	Array providing flow boundary conditions. i gives the flow boundary condition sequence number, marked 1, 2, etc. $j = 1$ gives the segment number imposed with the boundary condition; $j = 2$ gives the sequence number of the input flow time series.
`IPAIR(i,j)`	List of flow pairs; i = flowpair number, j = 1, 2 gives segment connected by the flow pair; $j = 3$ gives the type of the boundary condition or the flow pair type. If the type IPAIR(I,3) = 0, there is no flow; if = 1, it is a Manning's equation type; if = 3, it is a uniform flow type, with IPAIR(I,1) as segment for which the uniform flow is applied. IPAIR(I,3) = 4 and above gives various structure type boundary conditions.
`KA1`	Size of matrix A.
`KE`	maximum array size for the number of 2-D elements.
`KL`	Maximum array size for number of canal sections.
`KN`	Maximum array size for the number of 2-d nodes.
`KP`	Maximum array size for Number of canal joints.
`KR`	maximum array size for the number of canal flow pairs.
`LINK(i,j)`	$i = 1, 2, \ldots NJ$ gives the joint numbers; $j = 1$ gives the number of canal segment attached to the joint; $j = 2, 3, ..$ gives the specific segment numbers attached to the node.
`NJ`	Number of canal joints.
`NL`	Number of canal segments.
`QCBND(i)`	Flow boundary condition value provided for time series i.
`QPAIR(i)`	Discharge across flow pair i.
`XL(i)`	Length of canal segment (i).

Appendix E

UNET input data files used to carry out test 2

```
T1 Branch into two
T2 Example
T3 REACH 1
XK  99.9                              .5     0.5
NC   .02     .02     .02
PR ON
UB
X1   0.0       4       0   1000.  52800.  52800.  52800.
HY 1RM0.0
GR 49.04       0   29.04       0   29.04   1000.   49.04   1000.
X1   1.0       4       0   1000.      0.      0.      0.          -05.28
HY 1RM1.0
GR 49.04       0   29.04       0   29.04   1000.   49.04   1000.
DB     2       3
T1 Branch into two
T2 Example
T3 REACH 2
XK  99.9                              .5     0.5
UB     1
X1   1.0       4       0   2000. 105600. 105600. 105600.
HY 2RM1.0
GR 43.76       0   23.76       0   23.76   2000.   43.76   2000.
X1   3.0       4       0   2000.      0.      0.      0.          -21.12
HY 2RM3.0
GR 43.76       0   23.76       0   23.76   2000.   43.76   2000.
DB
T1 Branch into two
```

```
T2 Example
T3 REACH 3
XK  99.9                              .5     0.5
UB     1
X1   1.0       4       0    600. 158400. 158400. 158400.              0.
HY 3RM1.0
GR 43.76       0   23.76       0   23.76    600.   43.76    600.
X1   4.0       4       0    600.      0.      0.      0.          -23.76
HY 3RM4.0
GR 43.76       0   23.76       0   23.76    600.   43.76    600.
DB

EJ

------------------------------------------------------------
* Canal Branching
* Canal    PROBLEM
* TEST

JOB CONTROL
 T T  2MIN  0 -1  F 1. F T -1  2MIN

MAXINSTEPS=800
TIME WINDOW
 20OCT92 0000 20OCT92 2200

UPSTREAM FLOW AT REACH 1
1
7
0    20825.0
1    20825.0
1.333  50000.0
2    20825.0
6    20825.0
```

```
12   20825.0

22   20825.0

DOWNSTREAM STAGE HYDROGRAPH

2 5

0 6.6174

1 6.6174

2 6.6174

3 6.6174

5 6.6174

DOWNSTREAM STAGE HYDROGRAPH

3 5

0 3.9799

1 3.9799

2 3.9799

4 3.9799

5 3.9799

WRITE HYDROGRAPHS TO DSS

HBR.DSS

EJ
```